Univariate Time Series Analysis

Kevin Sheppard

Data

- All data from the Federal Reserve Economics Database (FRED) excluding the VWM return.
- Data from 1959 until end of 2020.

Series

- Industrial Production
- Curvature of the Yield Curve

$$(Y_{10} - Y_5) - (Y_5 - Y_1)$$

- Default Rate

$$Y_{Baa} - Y_{Aaa}$$

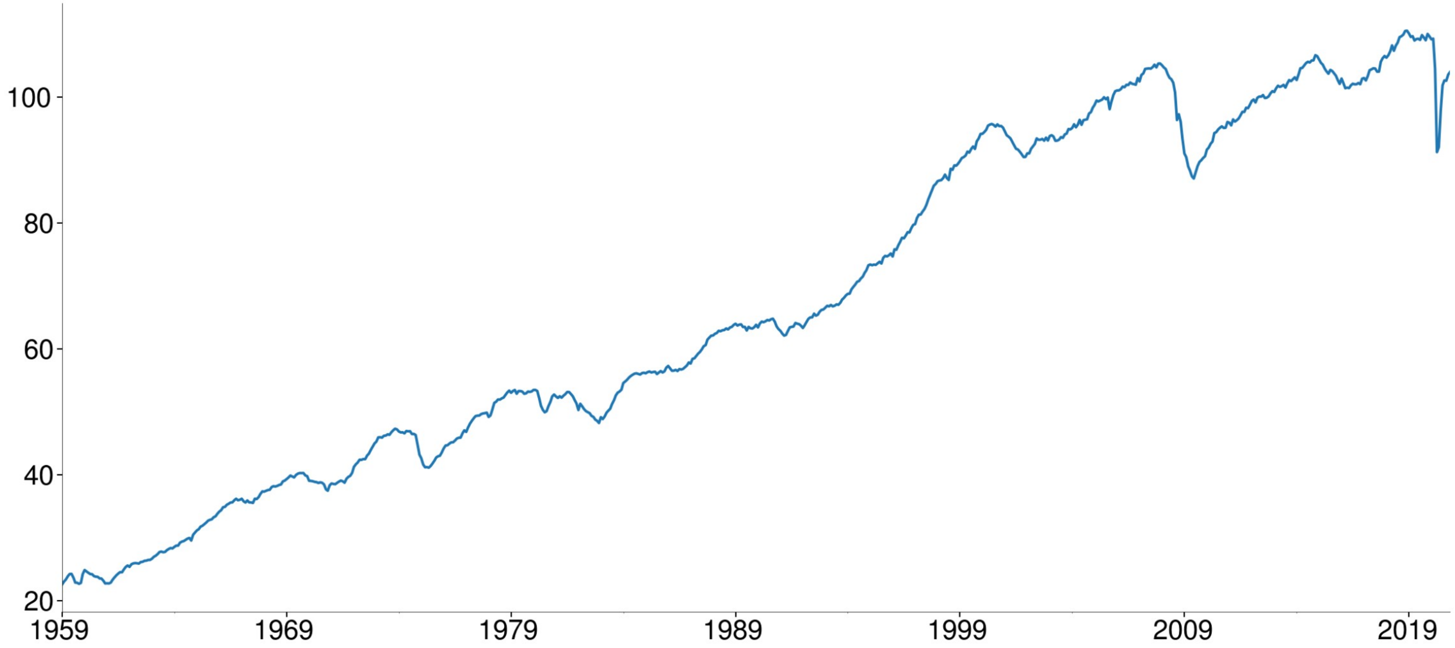
- US Construction workers
- Housing Starts (Not Seasonally Adjusted)
- Value-Weighted Market Return (CRSP)
- Simulated AR(2)

$$Y_t = 10 + 1.4Y_{t-1} - 0.5Y_{t-2} + \epsilon_t$$

Industrial Production Index

In [2]:

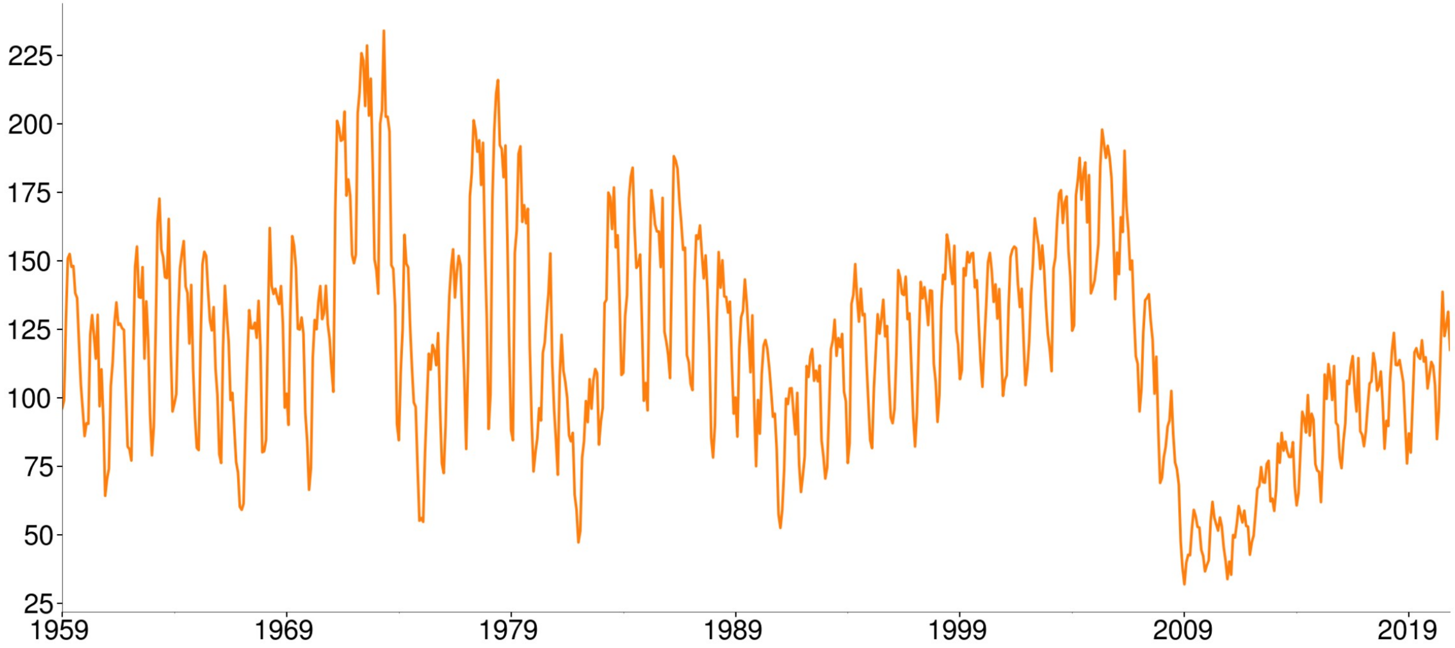
```
plot(orig.INDPRO)
```



Housing Starts (NSA)

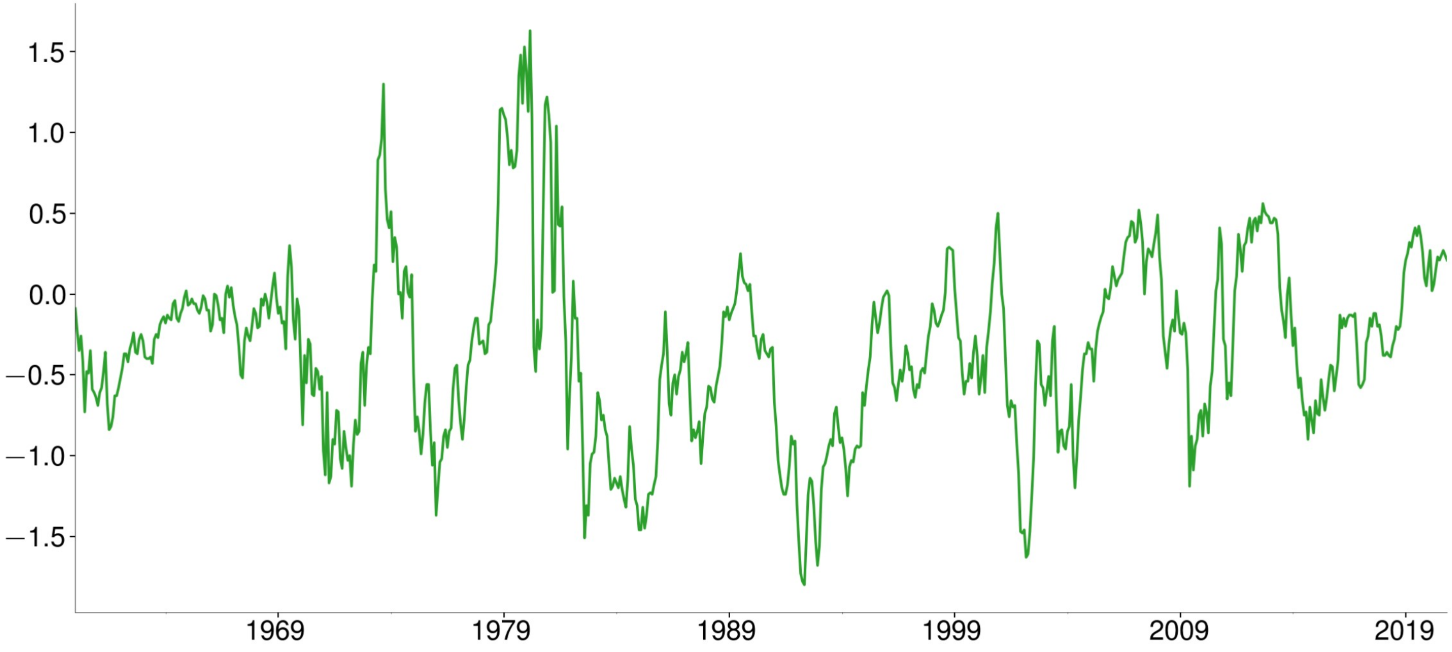
In [3]:

```
plot(orig.HOUSTNSA)
```



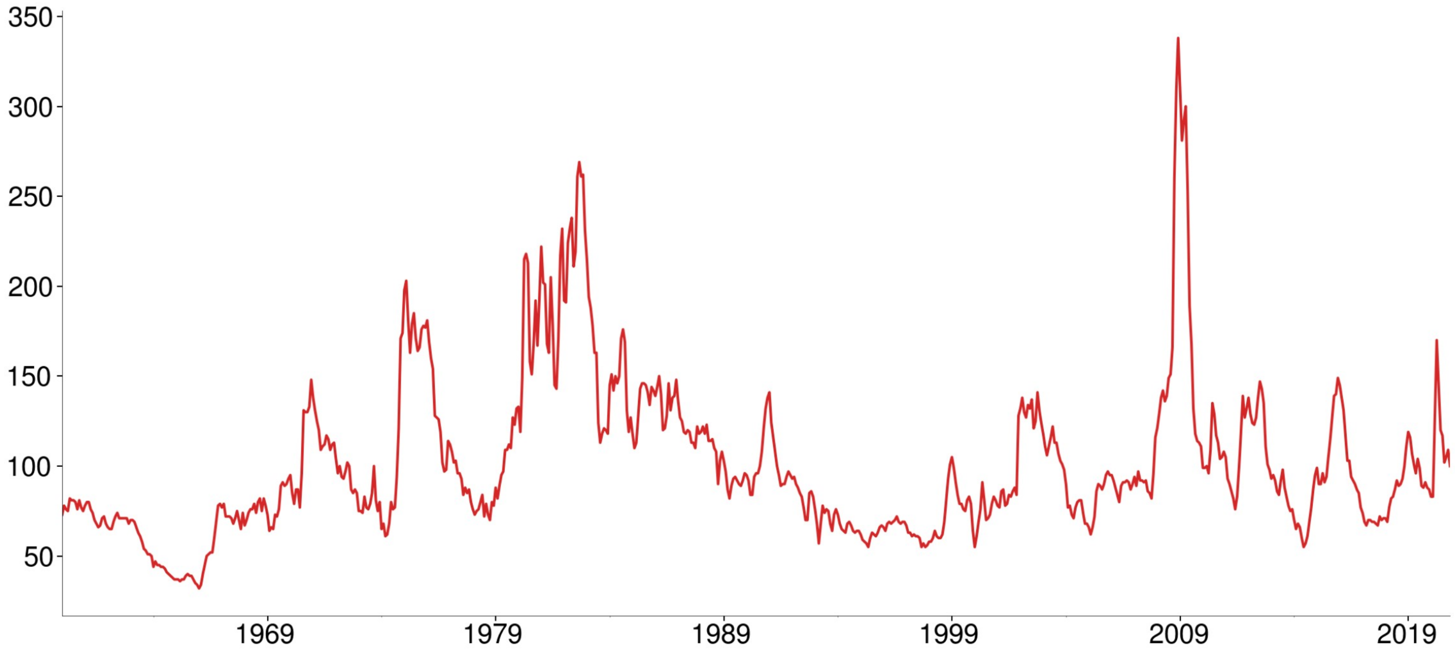
Curvature of the Yield Curve

In [4]:
```
plot(curve)
```



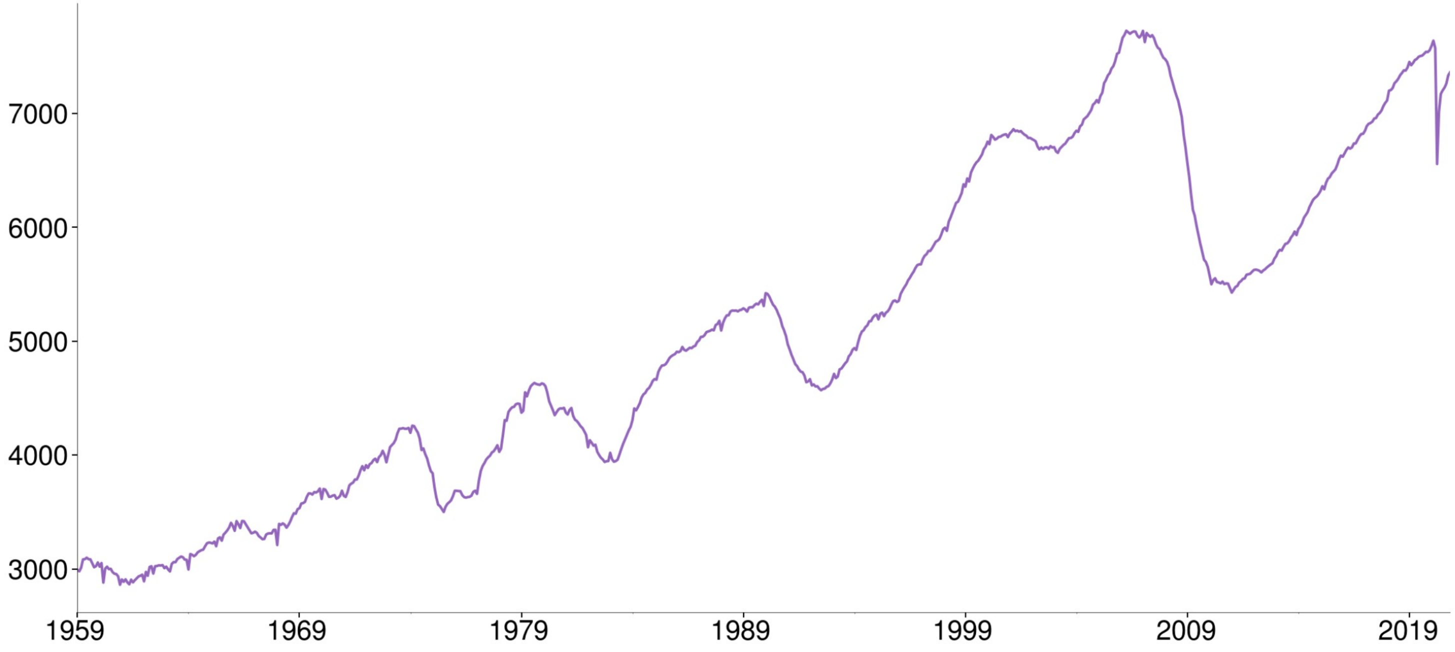
The Default Premium

In [5]:
```
plot(default)
```



US Construction Workers

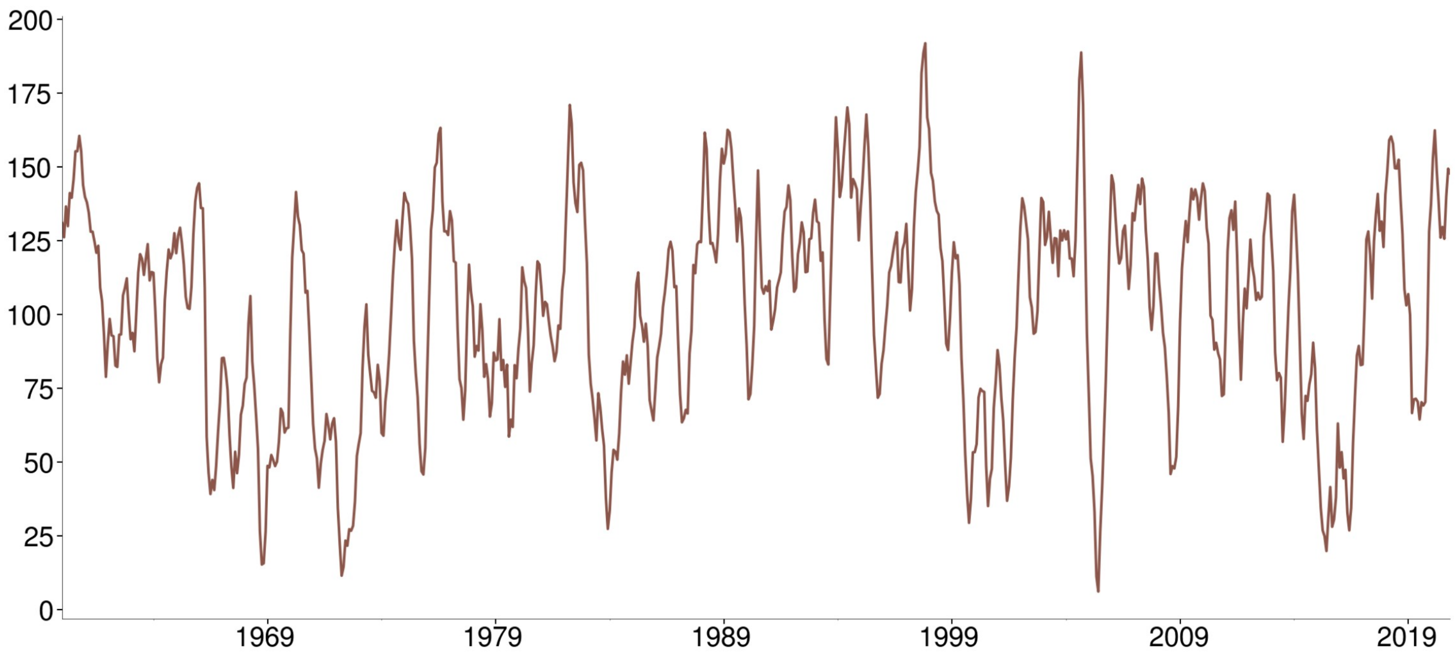
In [6]:
```
plot(orig.USCONS)
```



Simulated AR(2) Data

In [7]:

```
plot(sim)
```



Covariance Stationarity

A stochastic process Y_t is covariance stationary if

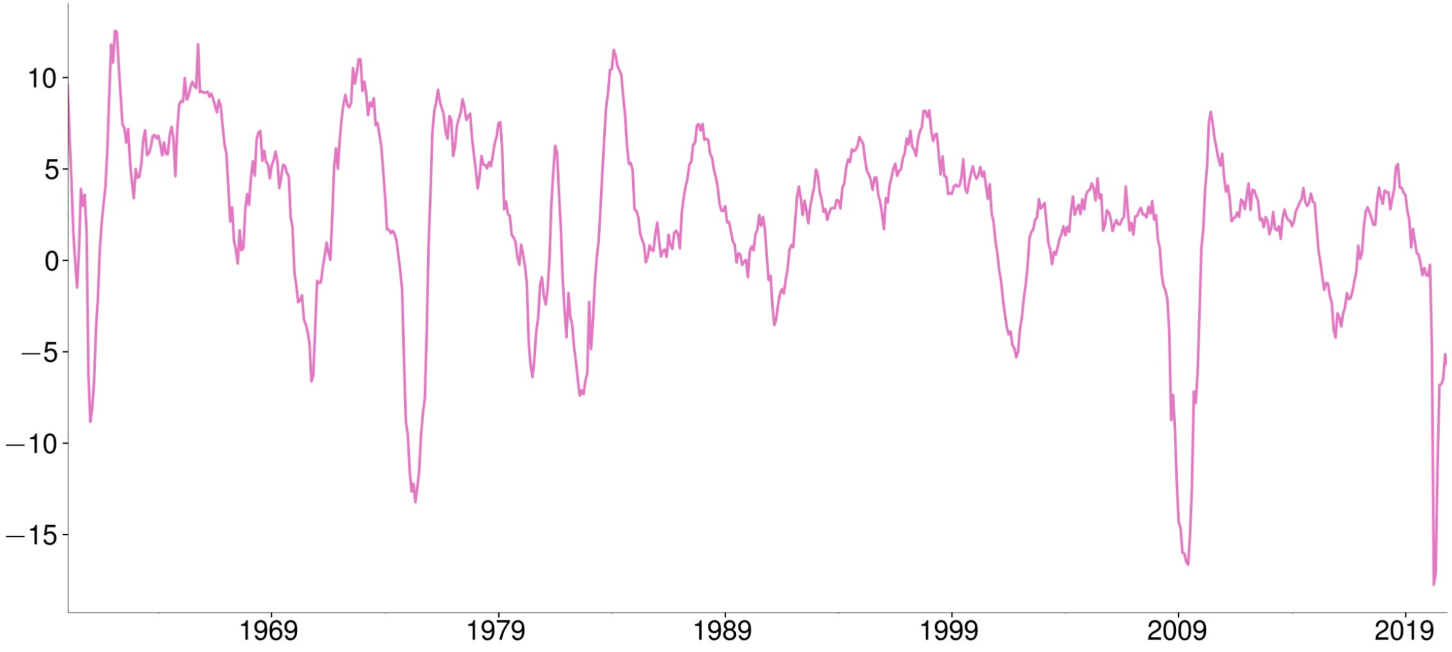
$$
\begin{aligned}
\mathrm{E}\left[Y_t\right] &= \mu \quad \text{for } t = 1, 2, \ldots \\
\mathrm{V}\left[Y_t\right] &= \sigma^2 < \infty \quad \text{for } t = 1, 2, \ldots \\
\mathrm{E}\left[(Y_t - \mu)(Y_{t-s} - \mu)\right] &= \gamma_s \quad \text{for } t = 1, 2, \ldots, s = 1, 2, \ldots, t-1
\end{aligned}
$$

Transformations

- Natural log ($\ln$): commonly applied to series that are non-negative
- Difference ($\Delta Y_t = Y_t - Y_{t-1}$): removes trends
- Log difference ($\Delta \ln Y_t$): growth rate
- Year-over-year Log Difference ($\Delta_1 2 \ln Y_t$): annual growth rate, removes *seasonality*

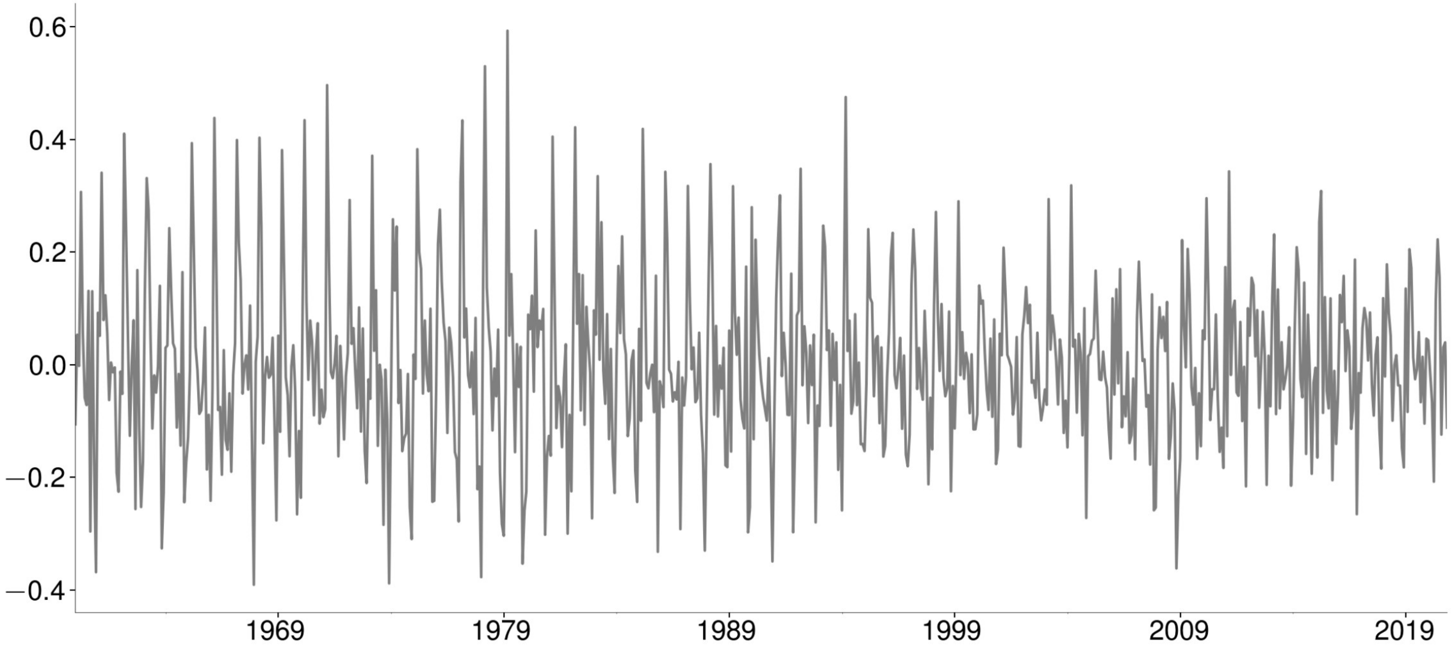
Year-over-Year Industrial Production Growth

In [8]:
```
plot(indpro)
```



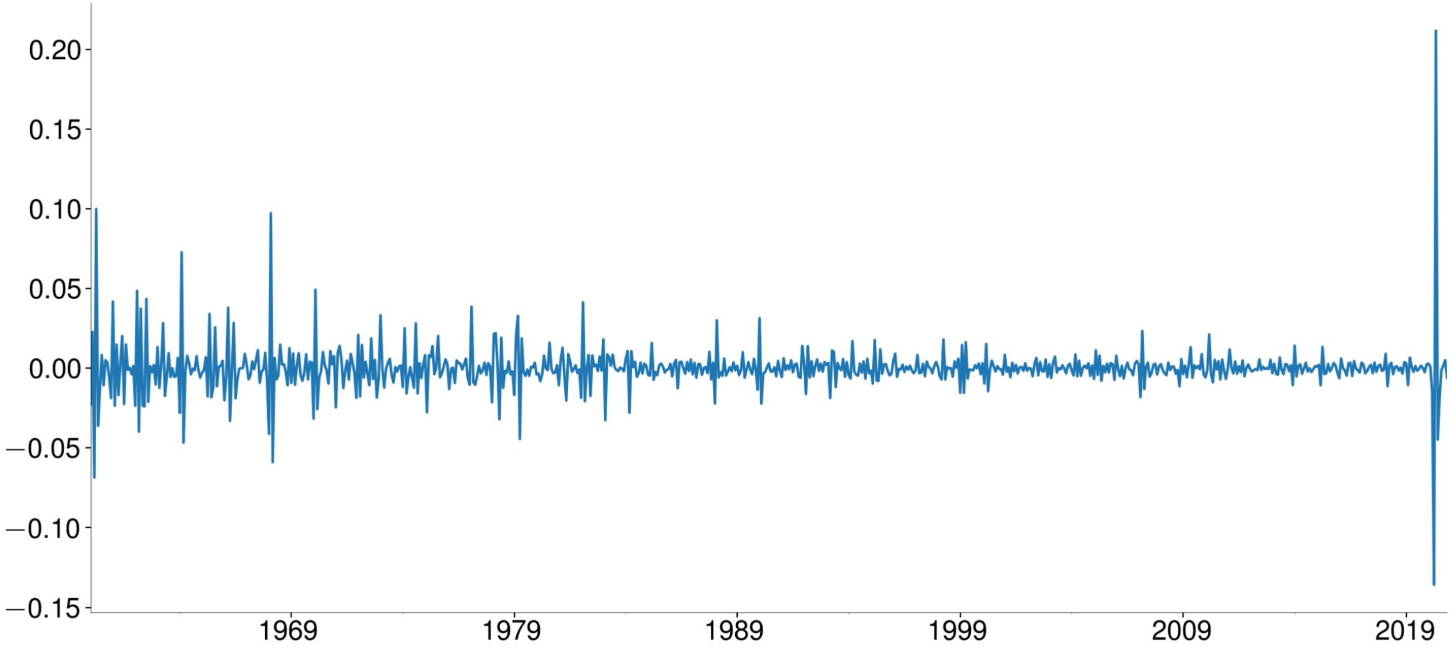
Housing Start Growth Rate

In [9]:
```
plot(housing)
```



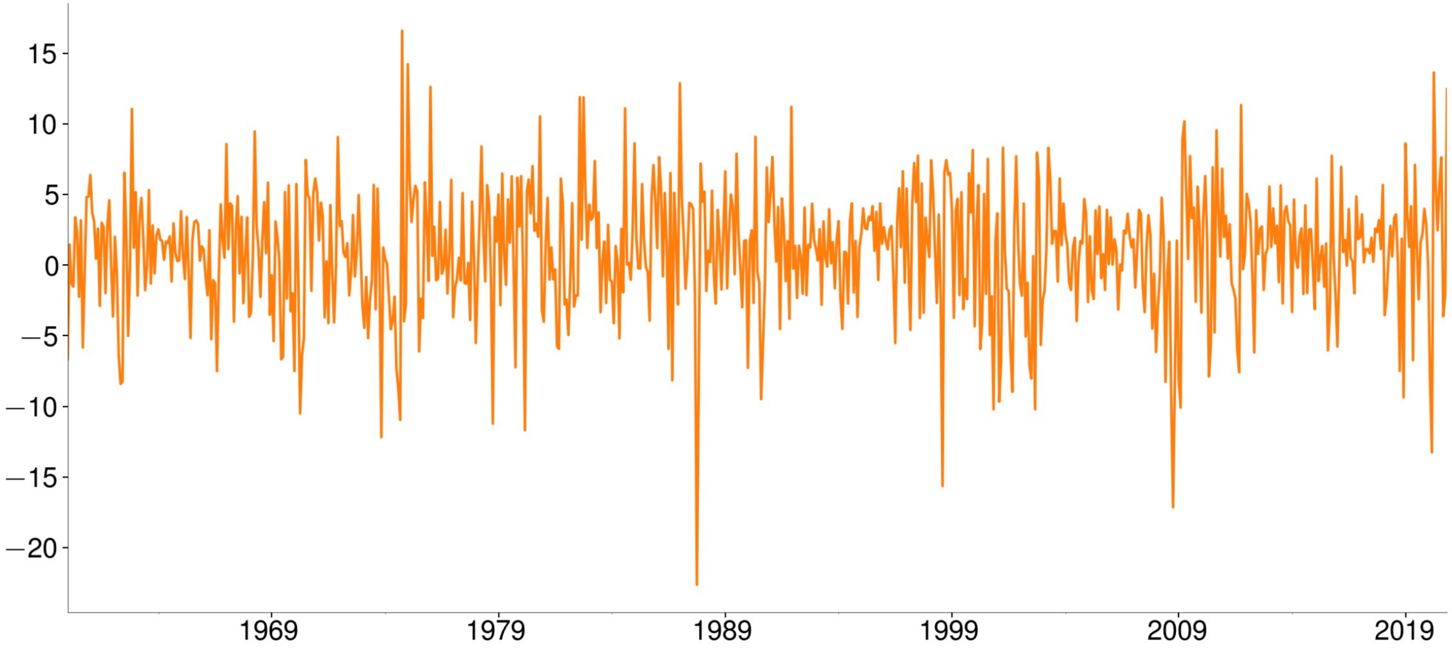
US Construction Workers Growth Rate

In [10]:
```
plot(uscons)
```



Value-Weighted Market

In [11]:
```
plot(vwm)
```



Identifying Candidate Models

- Exploit pattern and autocorrelation and partial autocorrelation
- Autocorrelation

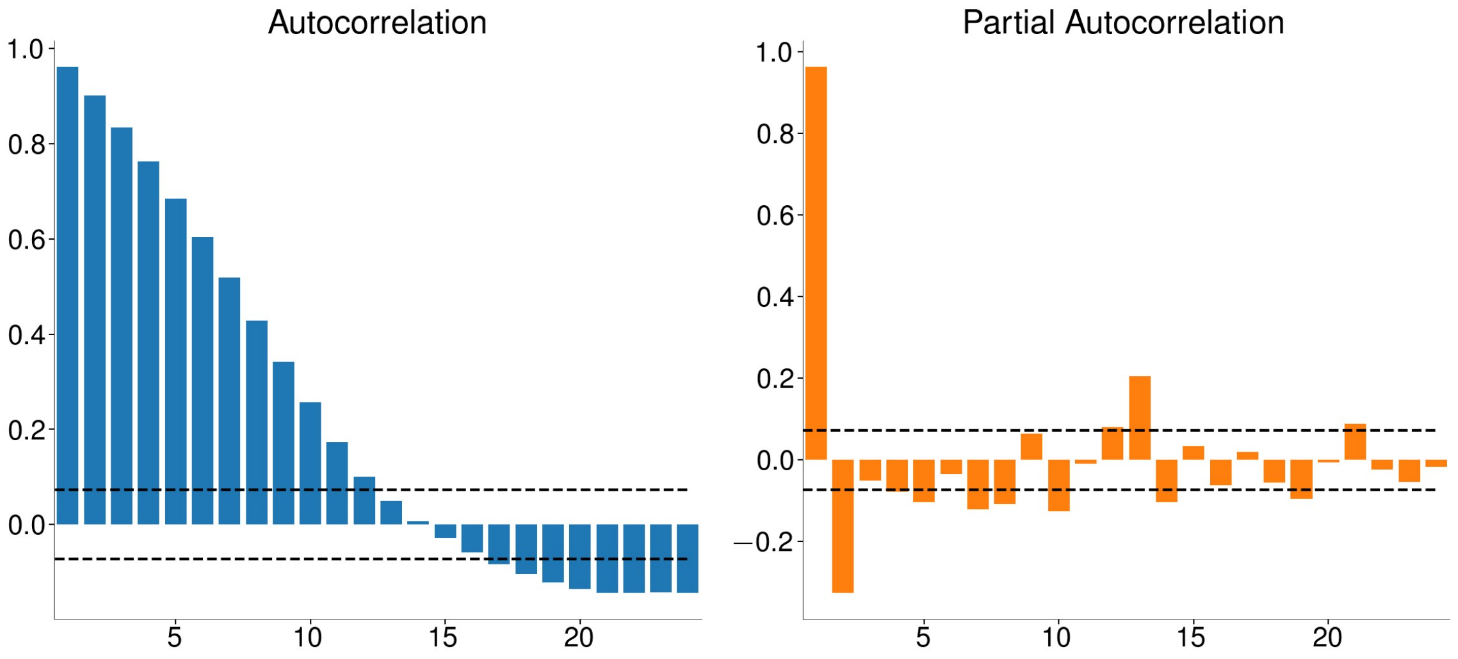
$$\mathrm{Corr}(Y_t, Y_{t-s}) = \frac{\gamma_s}{\gamma_0}$$

- Partial Autocorrelation

$$\mathrm{Corr}(Y_t, Y_{t-s} | Y_{t-1}, \ldots, Y_{t-s+1})$$

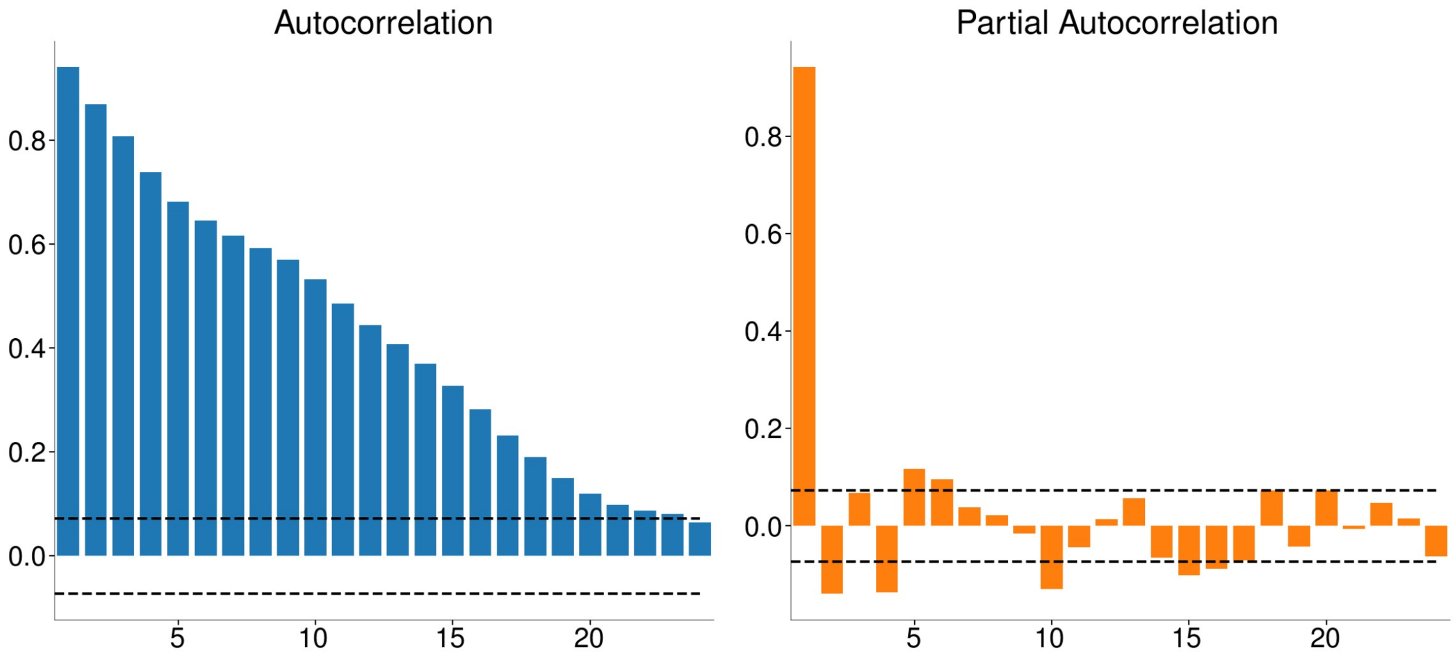
YoY Industrial Production Growth

In [13]:
```
acf_pacf_plot(indpro, 24)
```



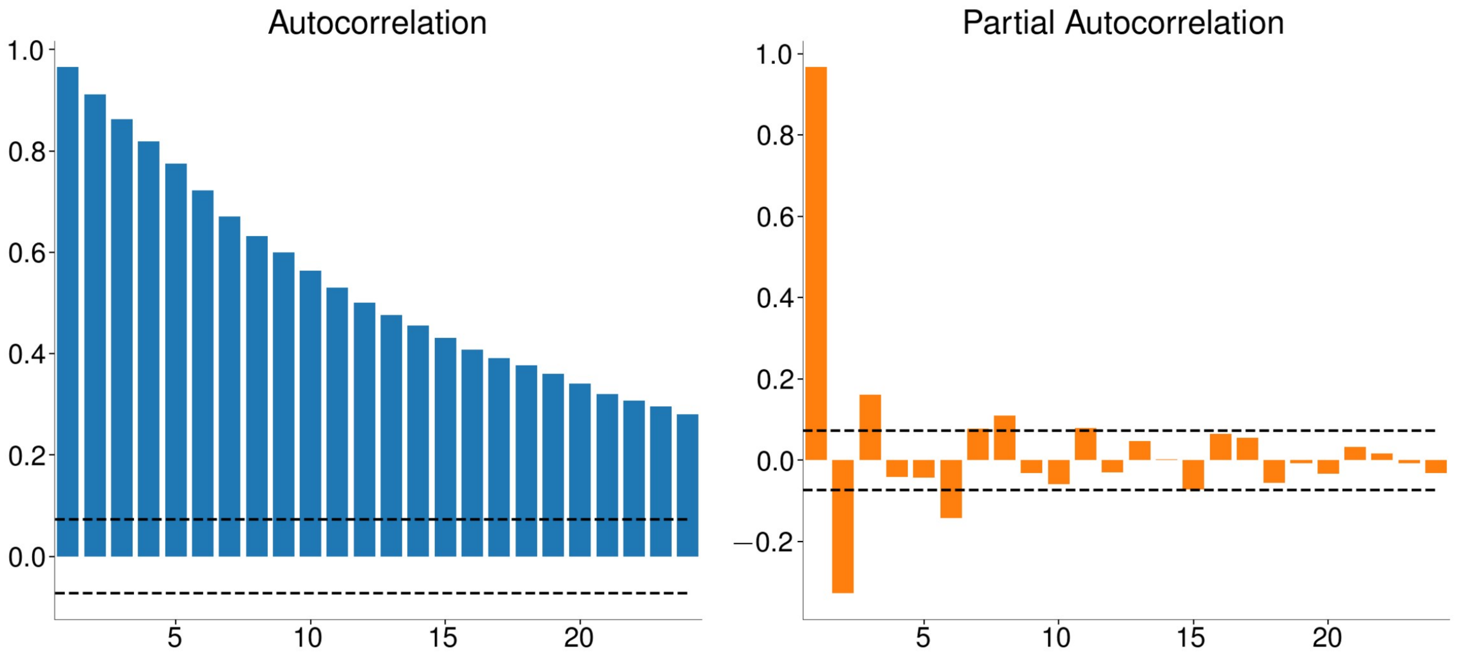
Curvature

In [14]:
```
acf_pacf_plot(curve, 24)
```



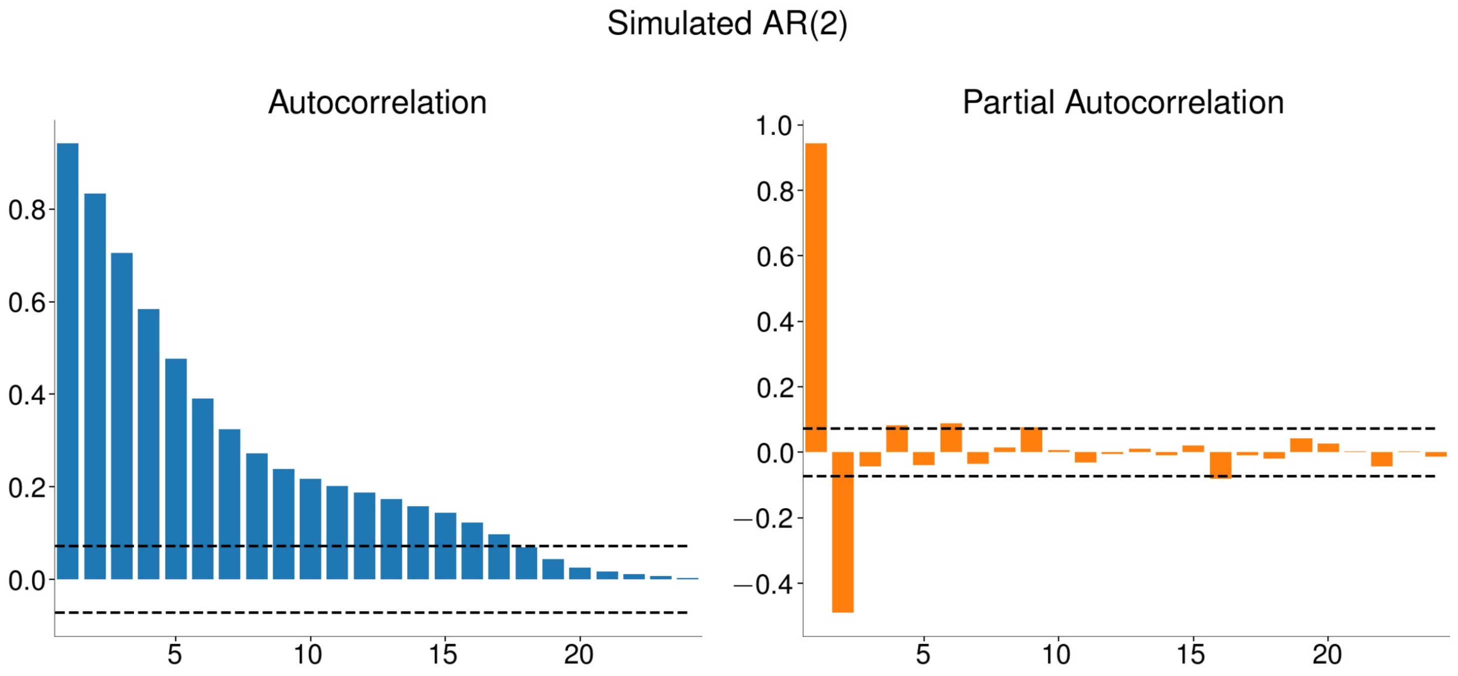
Default Premium

In [15]:
```
acf_pacf_plot(default, 24)
```



Simulated AR(2)

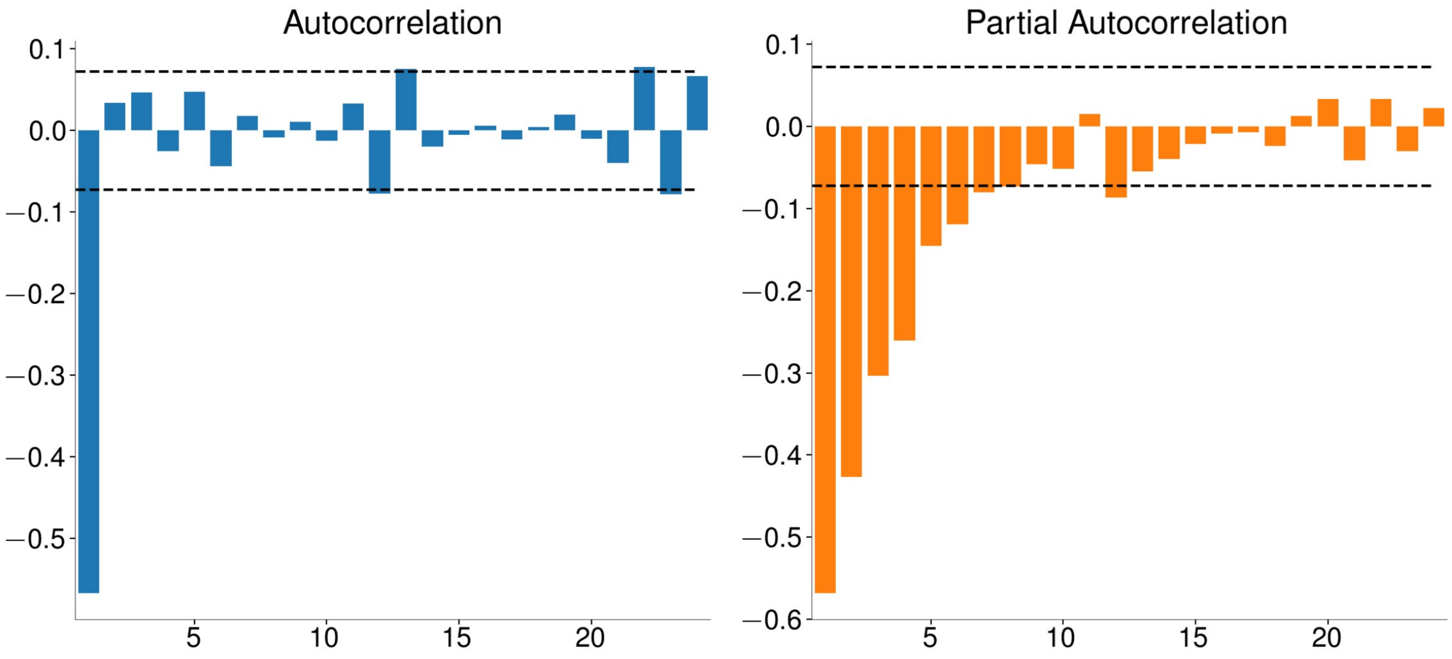
In [16]:
```
acf_pacf_plot(sim, 24, title="Simulated AR(2)")
```



Construction Worker Growth

In [17]:

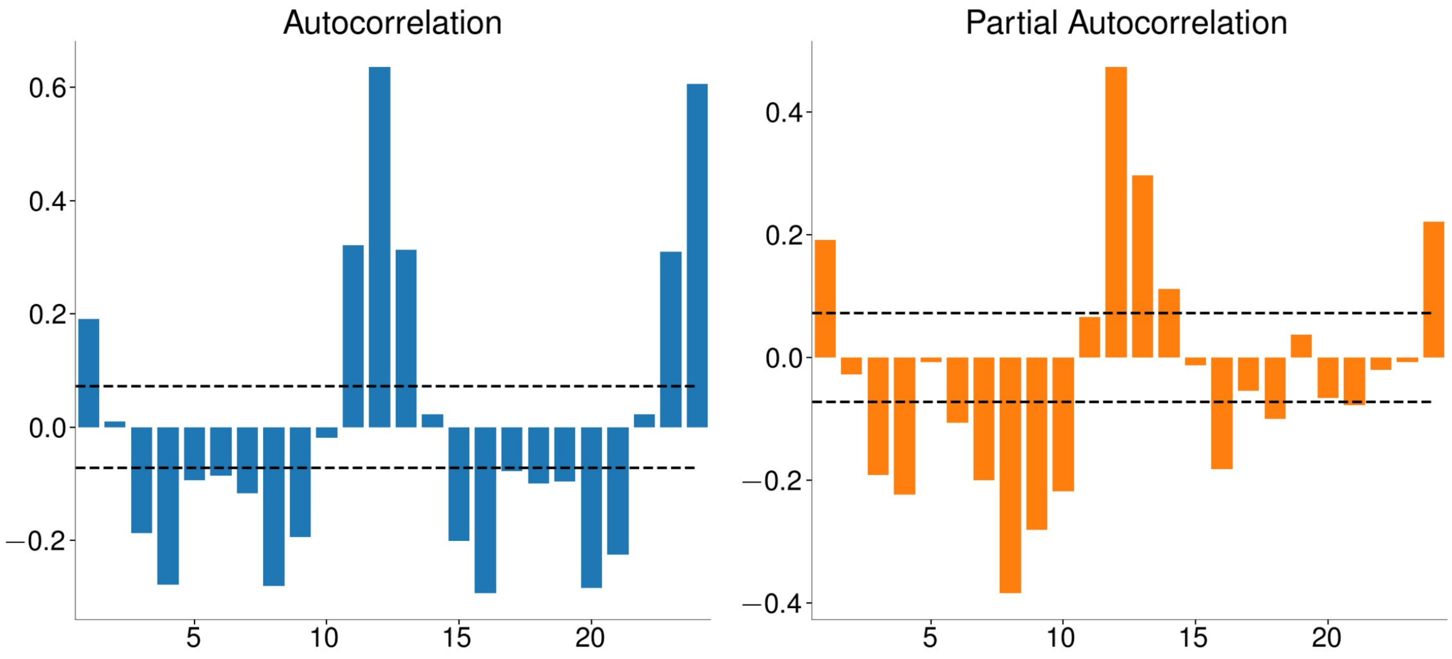
```
acf_pacf_plot(uscons, 24)
```



Housing Start Growth Rate

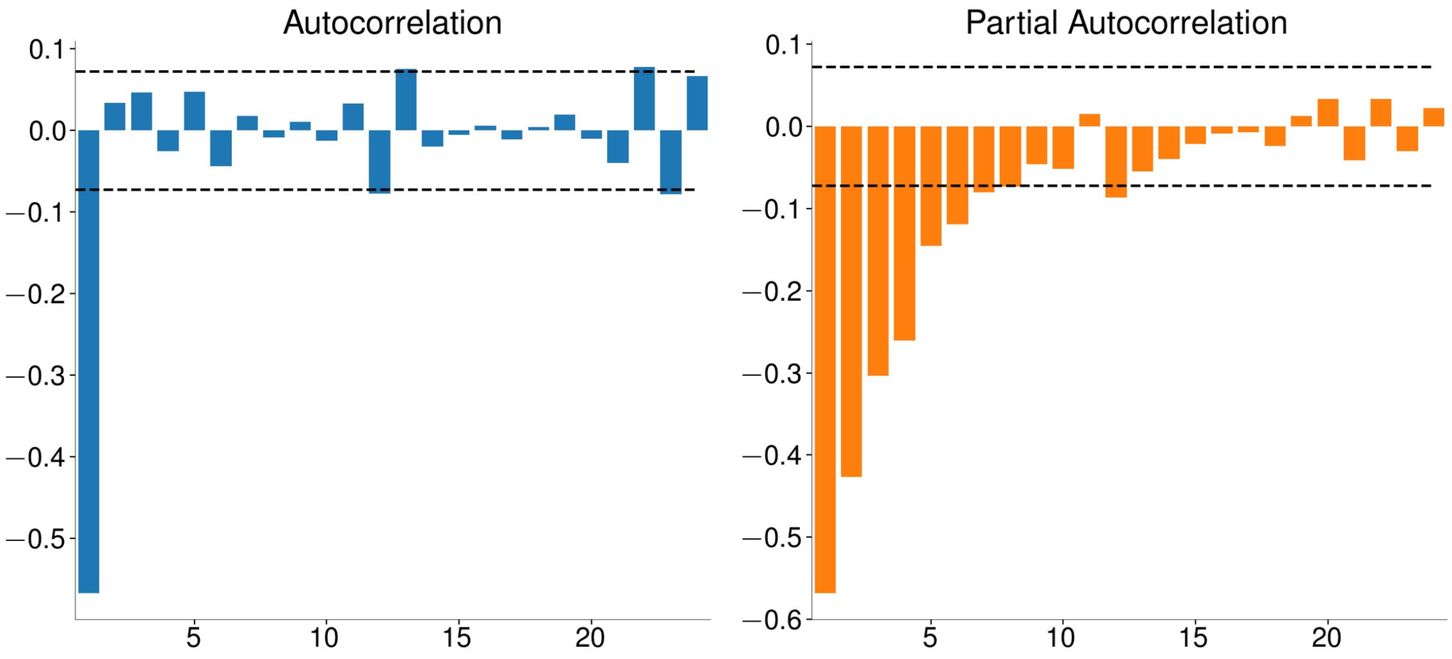
In [18]:

```
acf_pacf_plot(housing, 24)
```



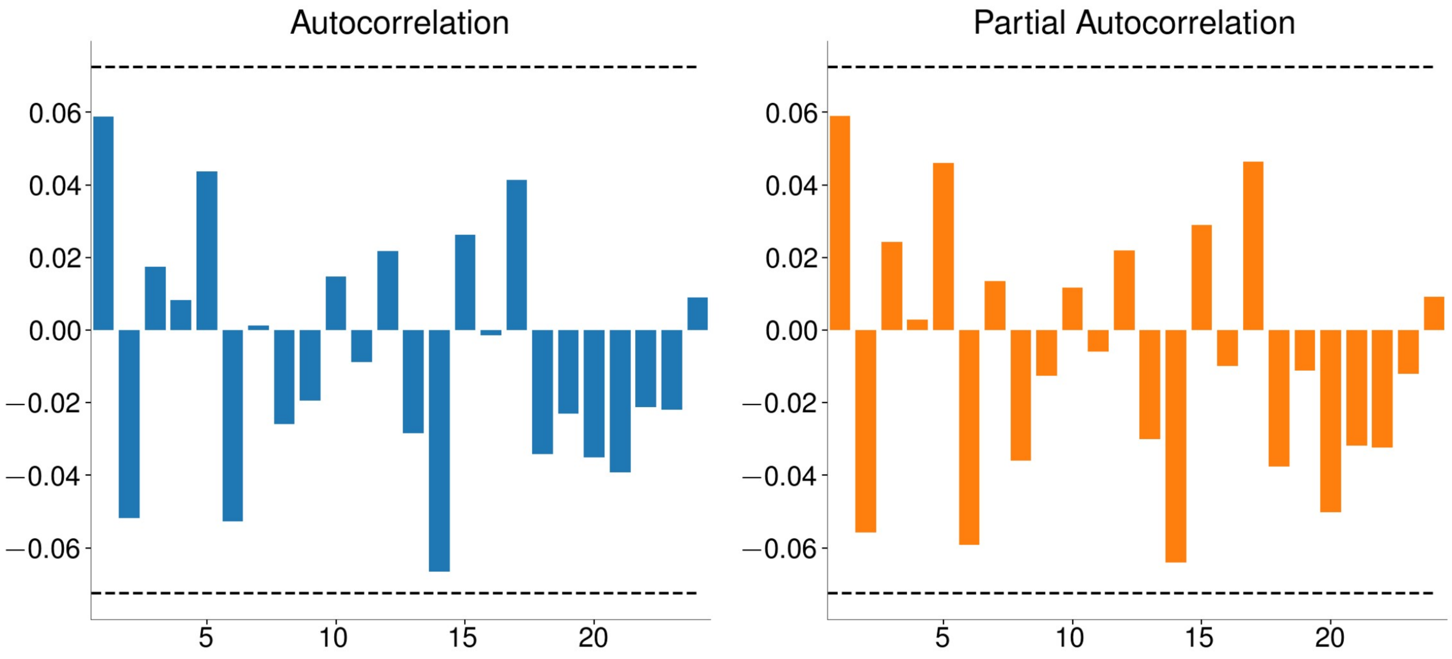
US Construction Worker Growth Rate

In [19]:
```
acf_pacf_plot(uscons, 24)
```



Value-Weighted Market

In [20]:
```
acf_pacf_plot(vwm, 24)
```



Fitting an initial model

- Begin with an AR(1) with a constant to Industrial Production Growth
$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \epsilon_t$$
- $\epsilon_t \sim WN\left(0, \sigma^2\right)$
- A process $\{\epsilon_t\}$ is known as white noise if
$$\begin{aligned} \mathrm{E}\left[\epsilon_t\right] &= 0 \qquad \text{for } t = 1, 2, \ldots \\ \mathrm{V}\left[\epsilon_t\right] &= \sigma^2 < \infty \qquad \text{for } t = 1, 2, \ldots \\ \mathrm{E}\left[\epsilon_t \epsilon_{t-j}\right] &= 0 \qquad \text{for } t = 1, 2, \ldots, \; j \neq 0 \end{aligned}$$
- Estimate parameters using Exact MLE
  - Discard $\max(P, Q)$ residuals
  - Model parameters include $\sigma^2 \equiv \mathrm{V}\left[\epsilon_t\right]$

Industrial Production Growth

In [22]:
```python
mod = SARIMAX(indpro, order=(1, 0, 0), trend="c")
res = mod.fit()
summary(res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.0780	0.046	1.708	0.088	-0.011	0.167
ar.L1	0.9673	0.007	132.070	0.000	0.953	0.982
sigma2	1.5703	0.027	58.347	0.000	1.518	1.623

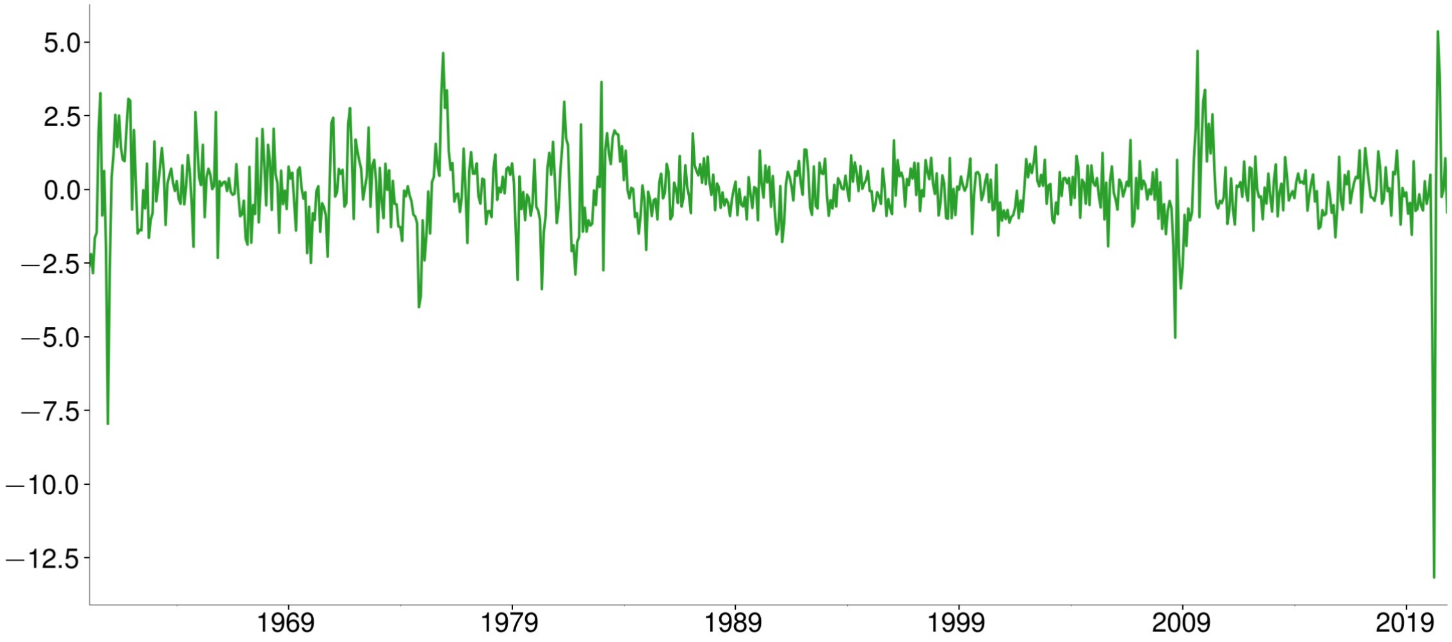
Diagnostics

- Plot residuals
- Residual Autocorrelations and Partial Autocorrelations
- Ljung-Box Q Test
- LM Test of Serial Correlation

Model Residuals

In [23]:

```
resid = res.resid.iloc[1:]
plot(resid)
```



Residual Autocorrelation

- Inference on autocorrelations

$$\begin{aligned} \mathrm{V}[\hat{\rho}_s] =& T^{-1} \text{ for } s = 1 \\ =& T^{-1}(1 + 2\sum_{j=1}^{s-1} \hat{\rho}_j^2) \text{ for } s > 1 \end{aligned}$$

- Inference on partial autocorrelations

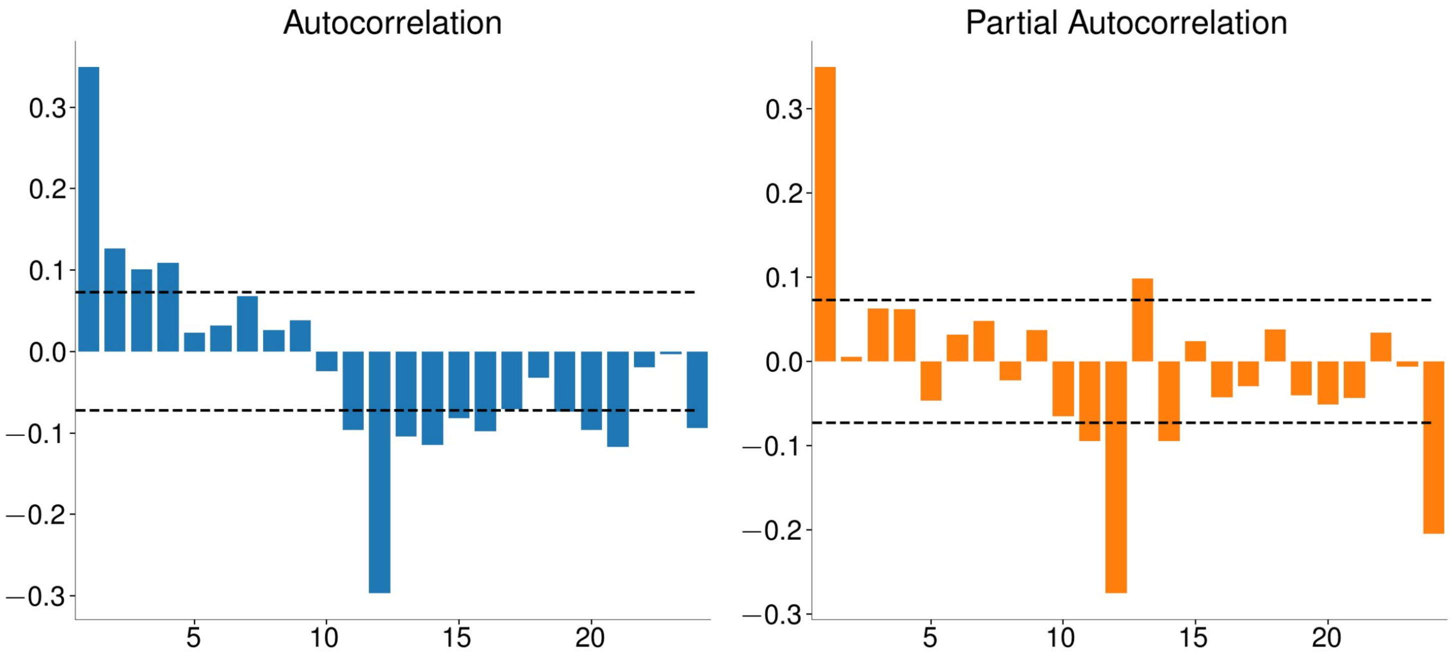
$$\mathrm{V}[\hat{\varphi}_s] \approx T^{-1}$$

- Confidence interval

$$1.96 \times \sqrt{V} \approx \frac{2}{\sqrt{T}}$$

Residual Autocorrelation

In [24]:
```
acf_pacf_plot(resid, 24)
```



Ljung-Box Q Statistic

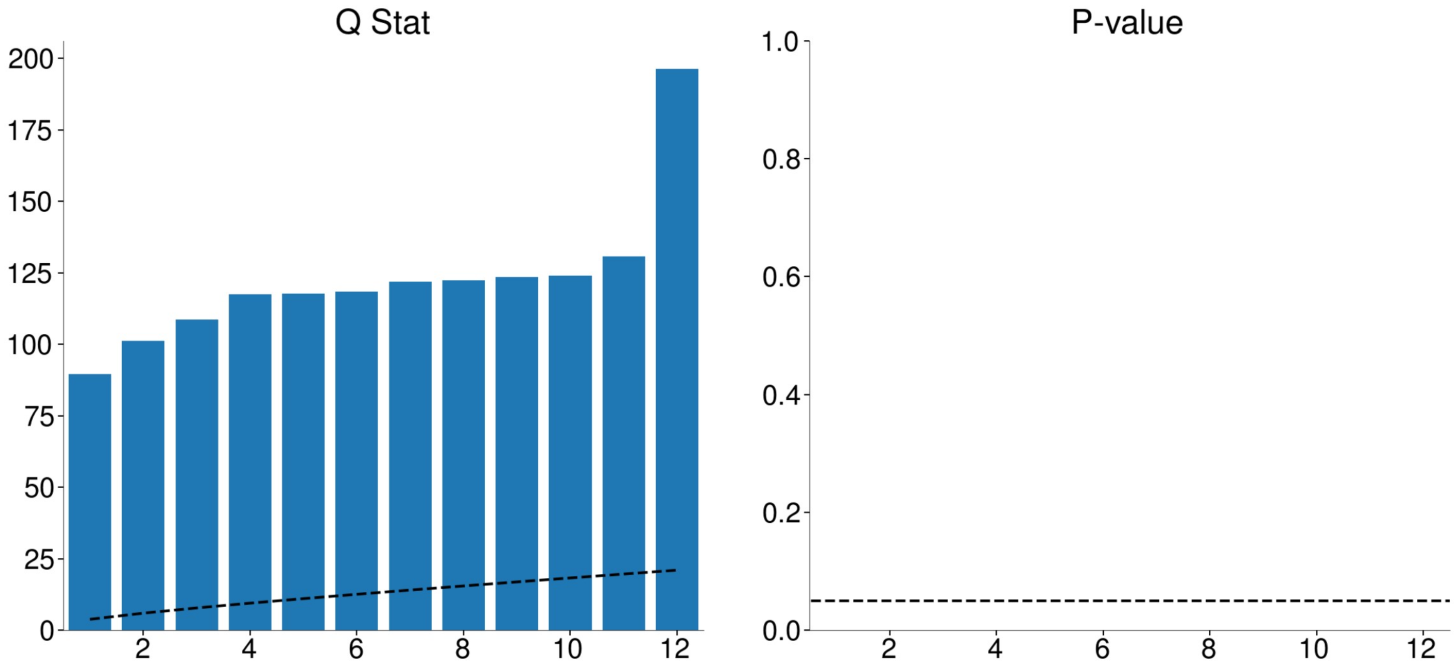
- Joint test of autocorrelations
- $H_0 : \rho_1 = \ldots = \rho_s = 0$

$$Q = T(T+2)\sum_{k=1}^{s} \frac{\hat{\rho}_k^2}{T-k} \sim \chi_s^2$$

- *Not* heteroskedasticity robust

Residual Q Statistic

In [26]:
```
lb_plot(resid, 12)
```



LM Test for Serial Correlation

Null $\mathrm{E}[Y_t^* Y_{t-j}^*] = 0$ for $1 \leq j \leq s \Longleftrightarrow \rho_j = 0$

The LM-test for serial correlation is constructed by defining the score vector $\mathbf{s}_t = Y_t^* \left[Y_{t-1}^* \; Y_{t-2}^* \; \ldots Y_{t-s}^*\right]'$

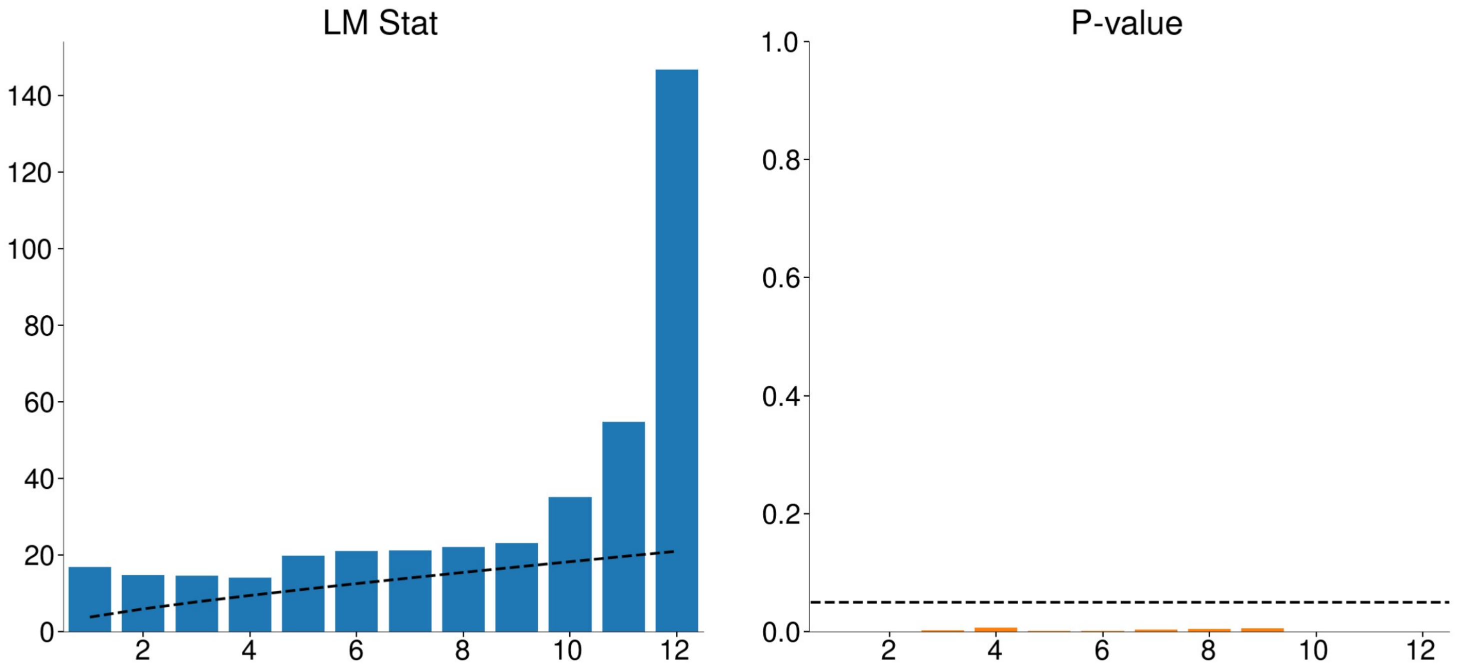
$$LM = T\bar{\mathbf{s}}'\hat{\mathbf{S}}^{-1}\bar{\mathbf{s}} \xrightarrow{d} \chi_s^2$$

- $\bar{\mathbf{s}} = T^{-1}\sum_{t=1}^{T} \mathbf{s}_t$
- $\hat{\mathbf{S}} = T^{-1}\sum_{t=1}^{T} \mathbf{s}_t \mathbf{s}_t'$
- $Y_t^* = Y_t - T^{-1} \sum_{t=1}^{T} Y_t$
- *Heteroskedasticity robust*

LM Stat and P-value

In [27]:

```
lm_plot(resid, 12)
```



Improving the Model

In [28]:
```
mod = SARIMAX(indpro, order=(2, 0, 0), trend="c")
res = mod.fit()
summary(res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	0.1106	0.045	2.453	0.014	0.022	0.199
ar.L1	1.3187	0.017	79.114	0.000	1.286	1.351
ar.L2	-0.3643	0.018	-20.624	0.000	-0.399	-0.330
sigma2	1.3635	0.028	48.775	0.000	1.309	1.418

Characteristic Roots

$$Y_t = \phi_0 + \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \epsilon_t$$

- Characteristic Equation
  $$z^2 - \phi_1 z - \phi_0 = 0$$
- Characteristic Roots
  $$(z - c_1)(z - c_2) = 0$$

Note: We will formalize testing for **unit roots** later

In [29]:
```
poly = np.array([1, -res.params["ar.L1"], -res.params["ar.L2"]])
poly
```

Out[29]:
```
array([ 1.        , -1.31873208,  0.3642808 ])
```

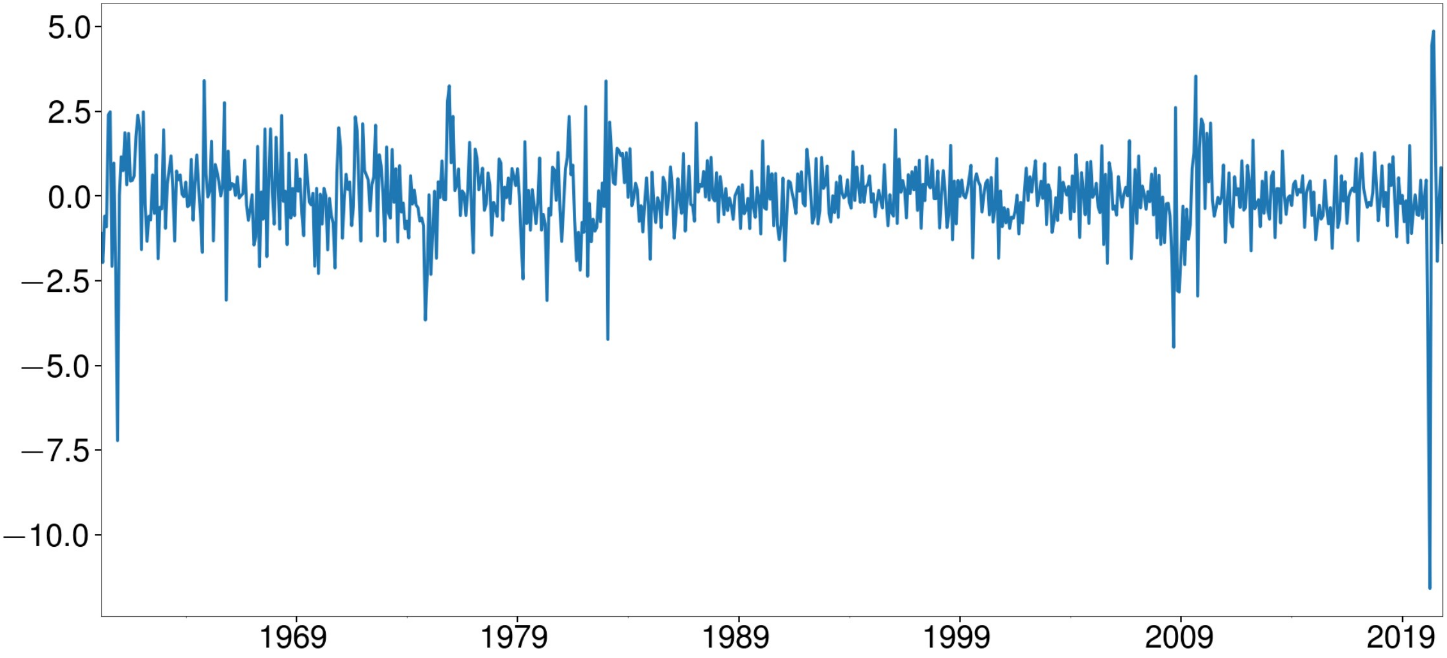
In [30]:
```
np.roots(poly)
```

Out[30]:
```
array([0.92485196, 0.39388012])
```

AR(2) Residuals

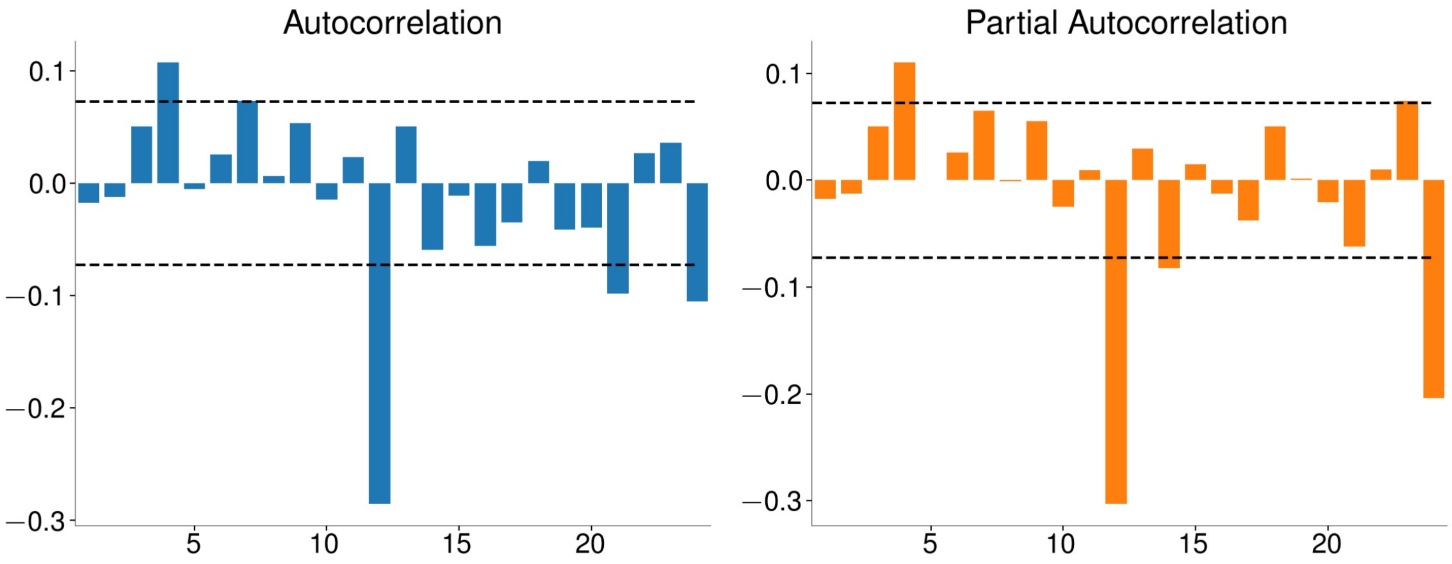
In [31]:

```
resid = res.resid.iloc[2:]
_ = resid.plot()
```



AR(2) Residual Autocorrelation

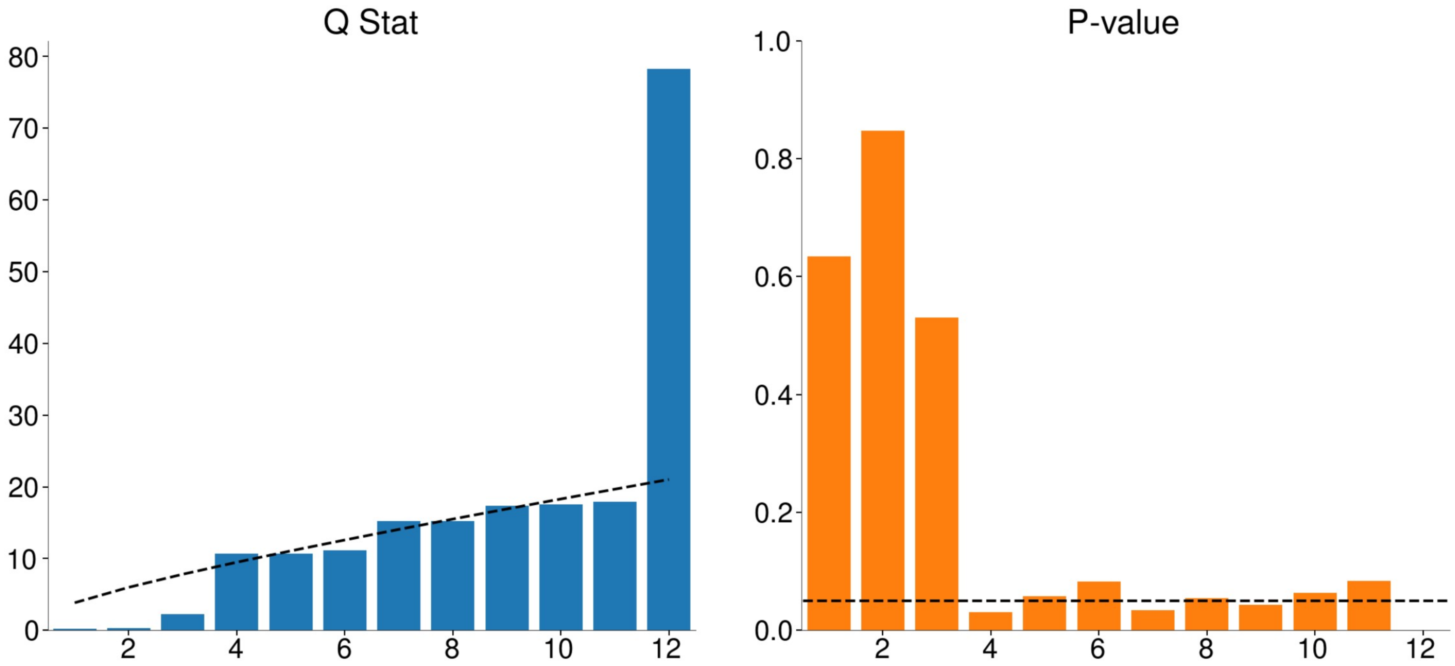
In [33]:
```
acf_pacf_plot(resid, 24)
```



AR(2) Q stat

In [35]:

```
lb_plot(resid, 12)
```



Modeling US Construction Worker Growth Rate

- Sample AC and PAC indicate an MA is needed

In [37]:
```
summary(res_ma1)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	2.063e-06	3.57e-05	0.058	0.954	-6.79e-05	7.21e-05
ma.L1	-0.9334	0.009	-102.611	0.000	-0.951	-0.916
sigma2	0.0001	1.33e-06	86.853	0.000	0.000	0.000

MA(2)

In [39]:

```
summary(res_ma2)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	8.191e-07	2.68e-05	0.031	0.976	-5.18e-05	5.34e-05
ma.L1	-1.0244	0.012	-88.805	0.000	-1.047	-1.002
ma.L2	0.0777	0.014	5.392	0.000	0.049	0.106
sigma2	0.0001	1.16e-06	91.835	0.000	0.000	0.000

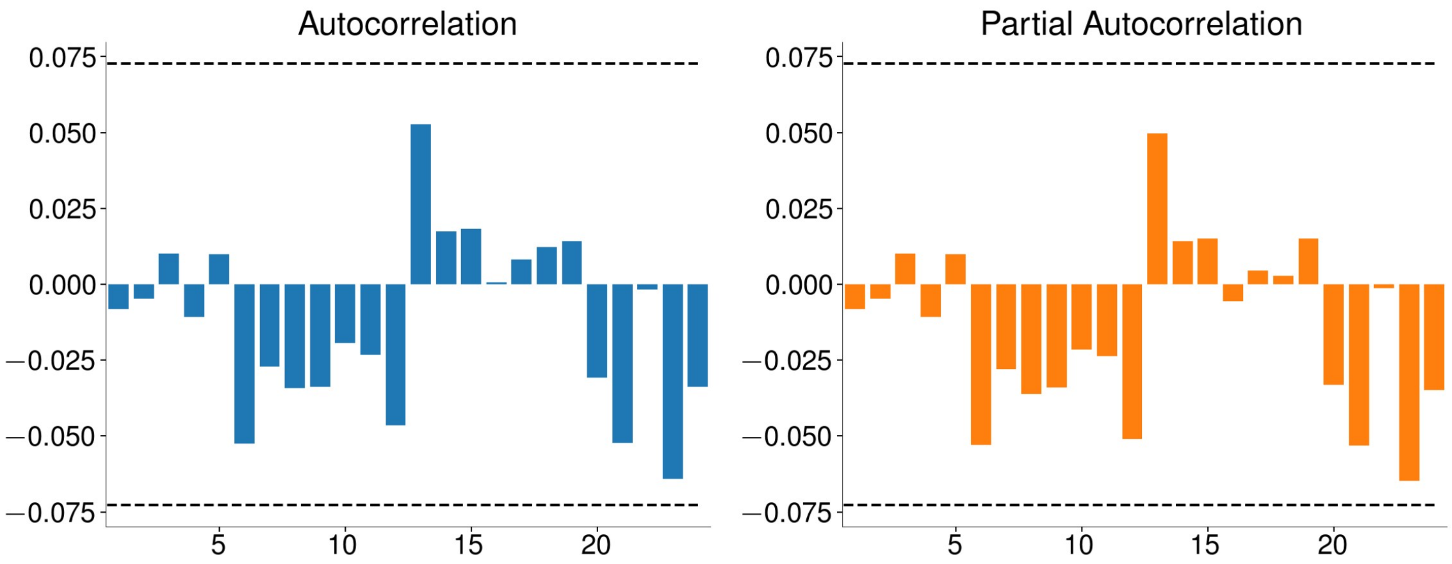
MA(3)

In [41]:
```
summary(res_ma3)
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	4.276e-06	8.31e-05	0.051	0.959	-0.000	0.000
ma.L1	-1.0516	0.012	-85.452	0.000	-1.076	-1.027
ma.L2	0.1228	0.021	5.933	0.000	0.082	0.163
ma.L3	0.0982	0.022	4.381	0.000	0.054	0.142
sigma2	0.0001	1.21e-06	88.482	0.000	0.000	0.000

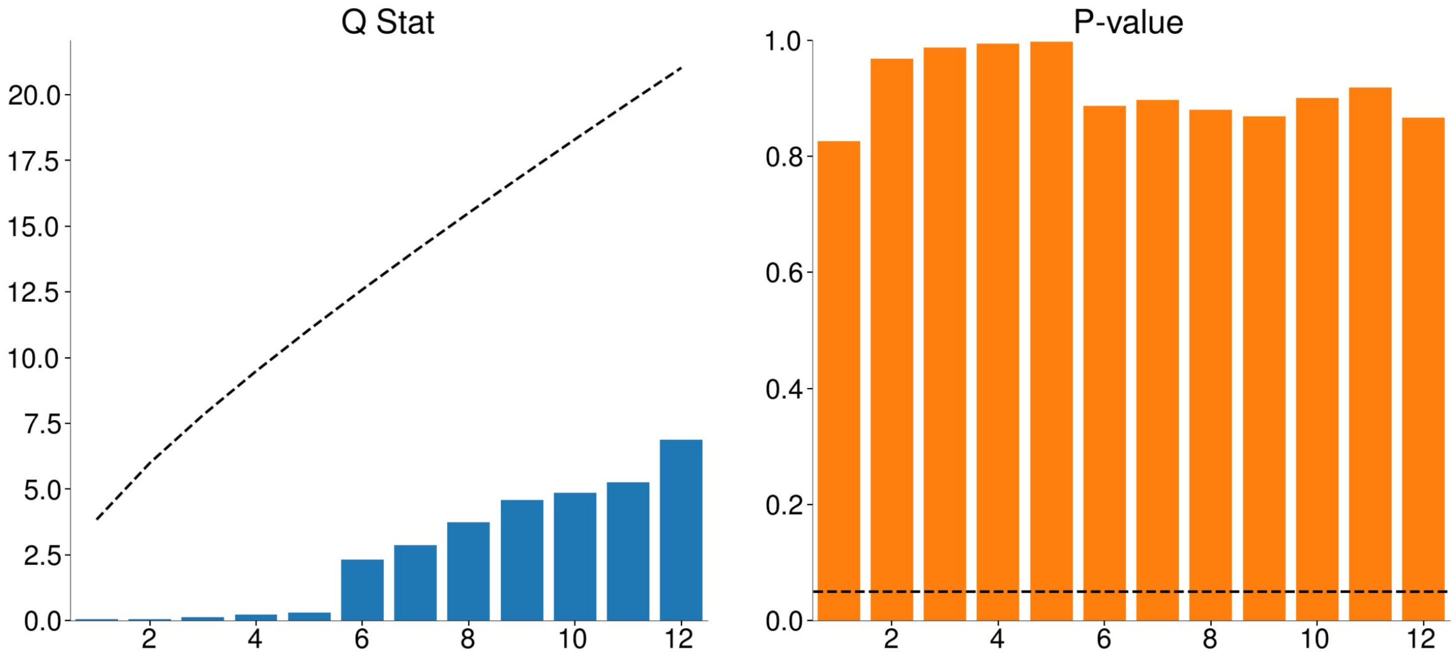
MA(3) Residual Analysis

In [43]:
```
resid = res_ma3.resid.iloc[3:]
acf_pacf_plot(resid, 24)
```



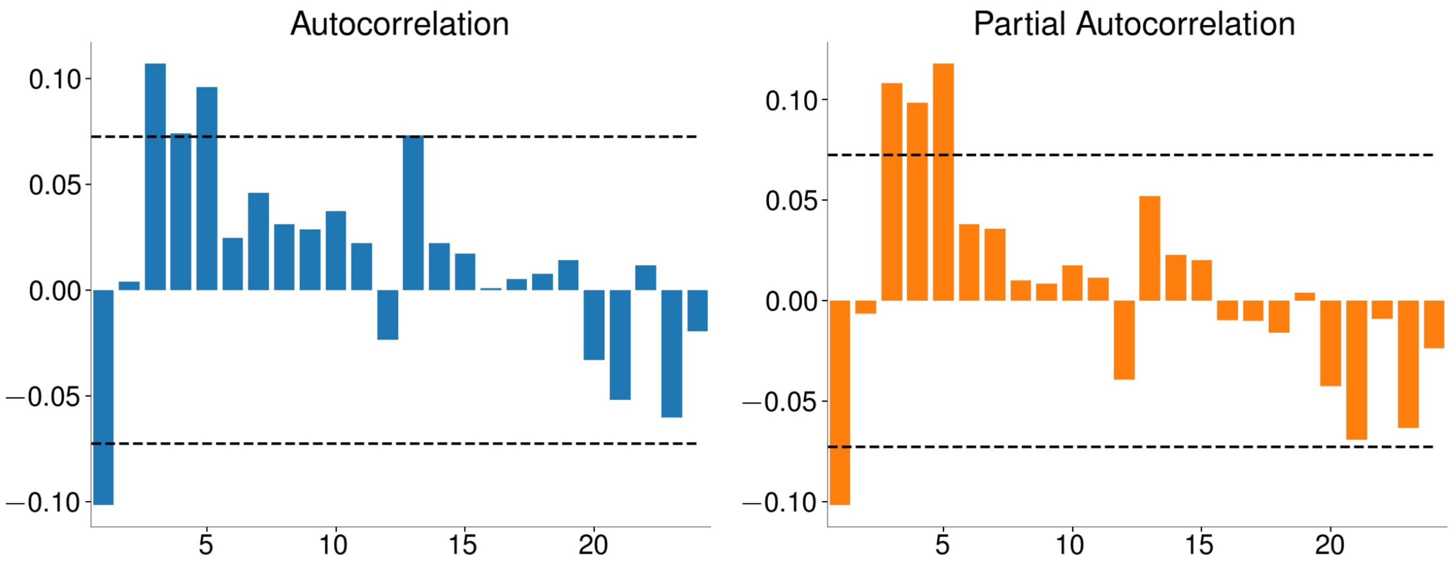
MA(3) Ljung-Box Q Statistic

In [45]:
```
lb_plot(resid, 12)
```



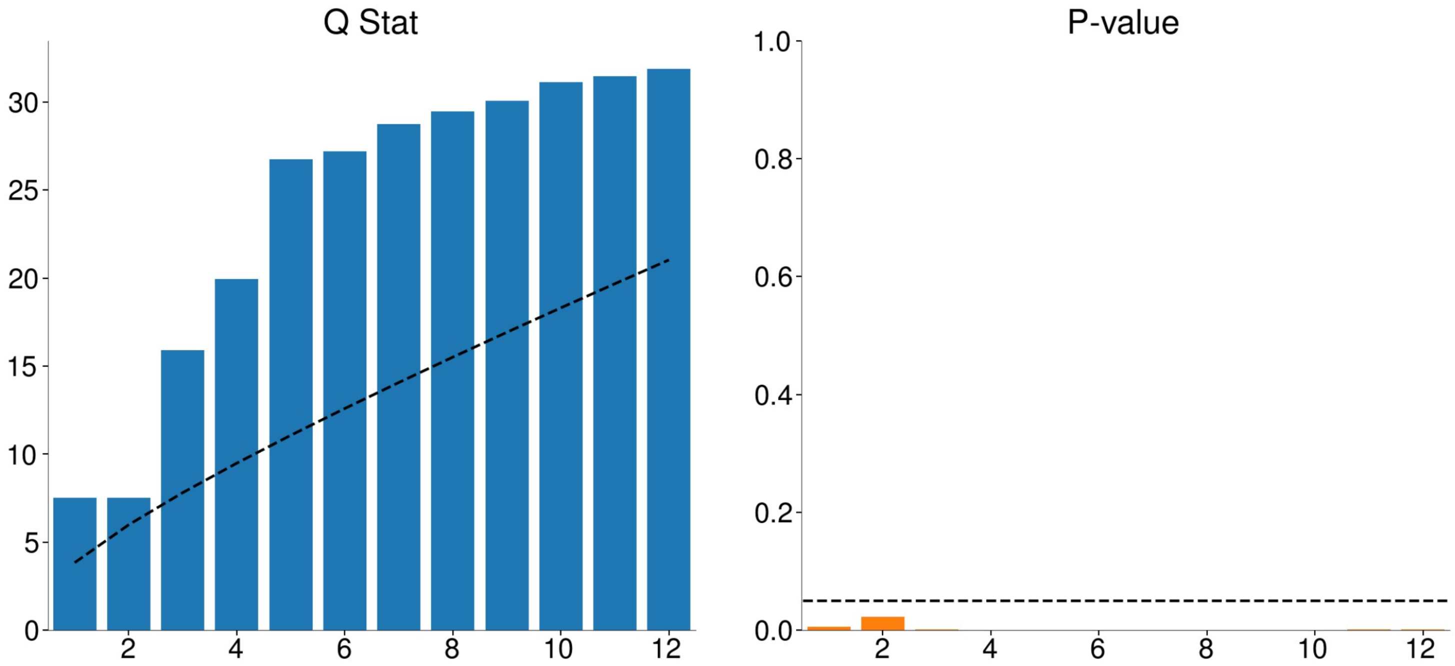
MA(1) Residual Analysis

In [47]:
```
resid = res_ma1.resid.iloc[3:]
acf_pacf_plot(resid, 24)
```



MA(1) Q Statistic

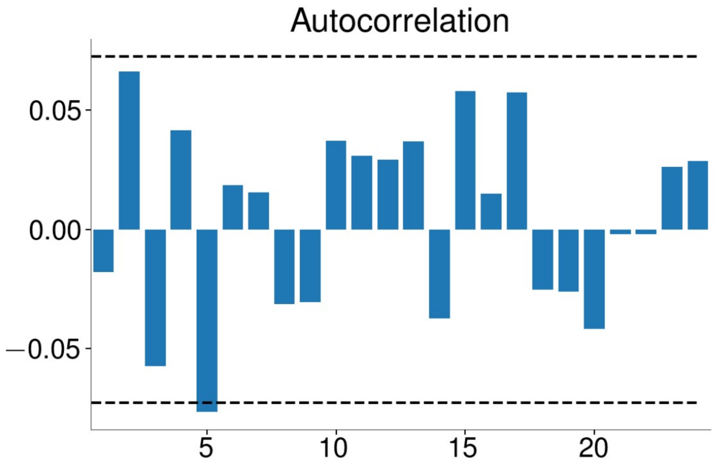
In [49]:
```
lb_plot(resid, 12)
```

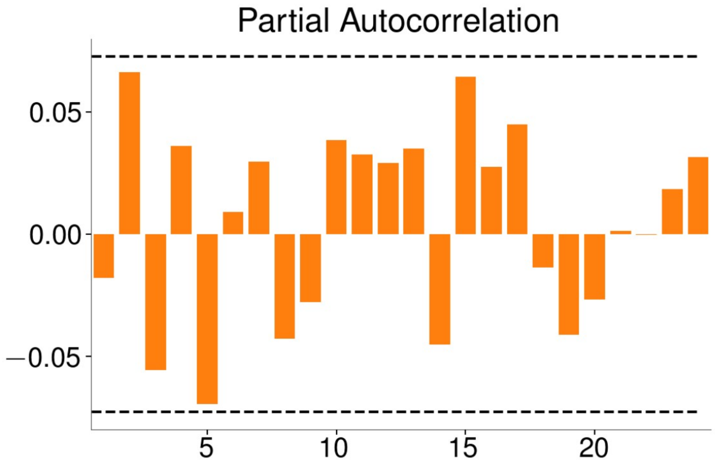


Correct Specification Benchmark AR(2)

In [51]:

```
res = SARIMAX(sim, order=(2, 0, 0), trend="c").fit()
resid = res.resid.iloc[2:]
acf_pacf_plot(resid, 24)
```

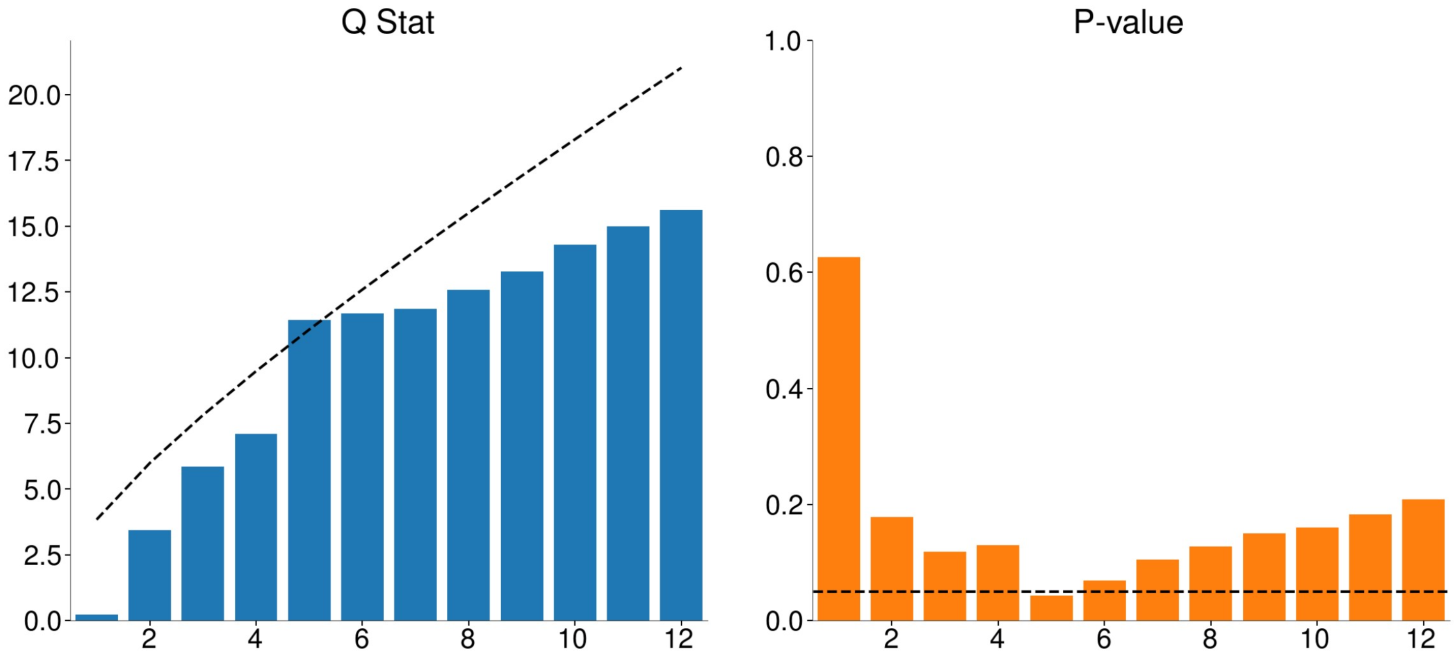




LB Q Stat on AR(2)

In [53]:

```
lb_plot(resid, 12)
```



StG Model Building

- Add either AR or MA
- Increase model complexity when statistically different using test size $\alpha \leq 5\%$
- Stop when no excluded lag is significant

Initial Models

In [55]:

```
summary(SARIMAX(default, order=(1, 0, 0), trend="c").fit())
summary(SARIMAX(default, order=(0, 0, 1), trend="c").fit())
```

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	3.4827	1.205	2.891	0.004	1.122	5.844
ar.L1	0.9652	0.007	139.901	0.000	0.952	0.979
sigma2	126.9150	2.638	48.103	0.000	121.744	132.086

	coef	std err	z	P>|z|	[0.025	0.975]
intercept	101.2112	2.446	41.378	0.000	96.417	106.005
ma.L1	0.9218	0.008	118.011	0.000	0.906	0.937
sigma2	586.9873	22.526	26.059	0.000	542.838	631.137

Extending the AR(1)

In [56]:

```
summary(SARIMAX(default, order=(2, 0, 0)).fit())
summary(SARIMAX(default, order=(1, 0, 1)).fit())
```

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	1.2913	0.021	62.923	0.000	1.251	1.332
ar.L2	-0.2985	0.021	-14.193	0.000	-0.340	-0.257
sigma2	117.3203	2.426	48.366	0.000	112.566	122.075

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9906	0.003	293.863	0.000	0.984	0.997
ma.L1	0.3752	0.021	17.643	0.000	0.334	0.417
sigma2	114.1891	2.392	47.736	0.000	109.501	118.877

Extending the ARMA(1,1)

In [57]:

```
summary(SARIMAX(default, order=(2, 0, 1)).fit())
summary(SARIMAX(default, order=(1, 0, 2)).fit())
```

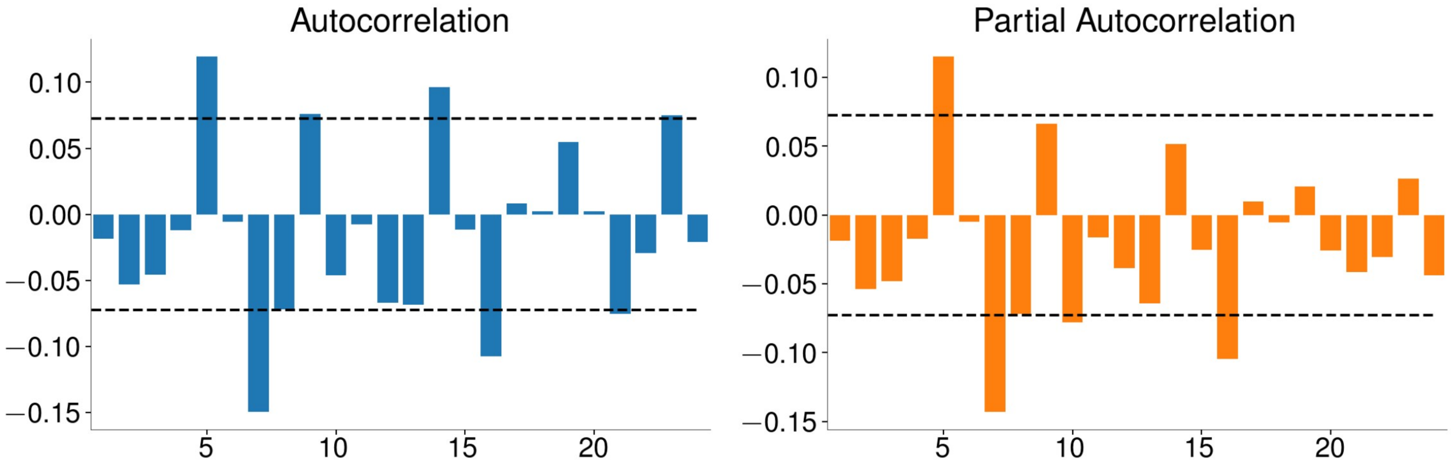
	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.8821	0.061	14.522	0.000	0.763	1.001
ar.L2	0.1082	0.061	1.783	0.075	-0.011	0.227
ma.L1	0.4687	0.054	8.611	0.000	0.362	0.575
sigma2	113.9984	2.389	47.715	0.000	109.316	118.681

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.9915	0.003	303.425	0.000	0.985	0.998
ma.L1	0.3554	0.023	15.541	0.000	0.311	0.400
ma.L2	-0.0490	0.023	-2.150	0.032	-0.094	-0.004
sigma2	113.9618	2.381	47.860	0.000	109.295	118.629

Selected Model Diagnostics

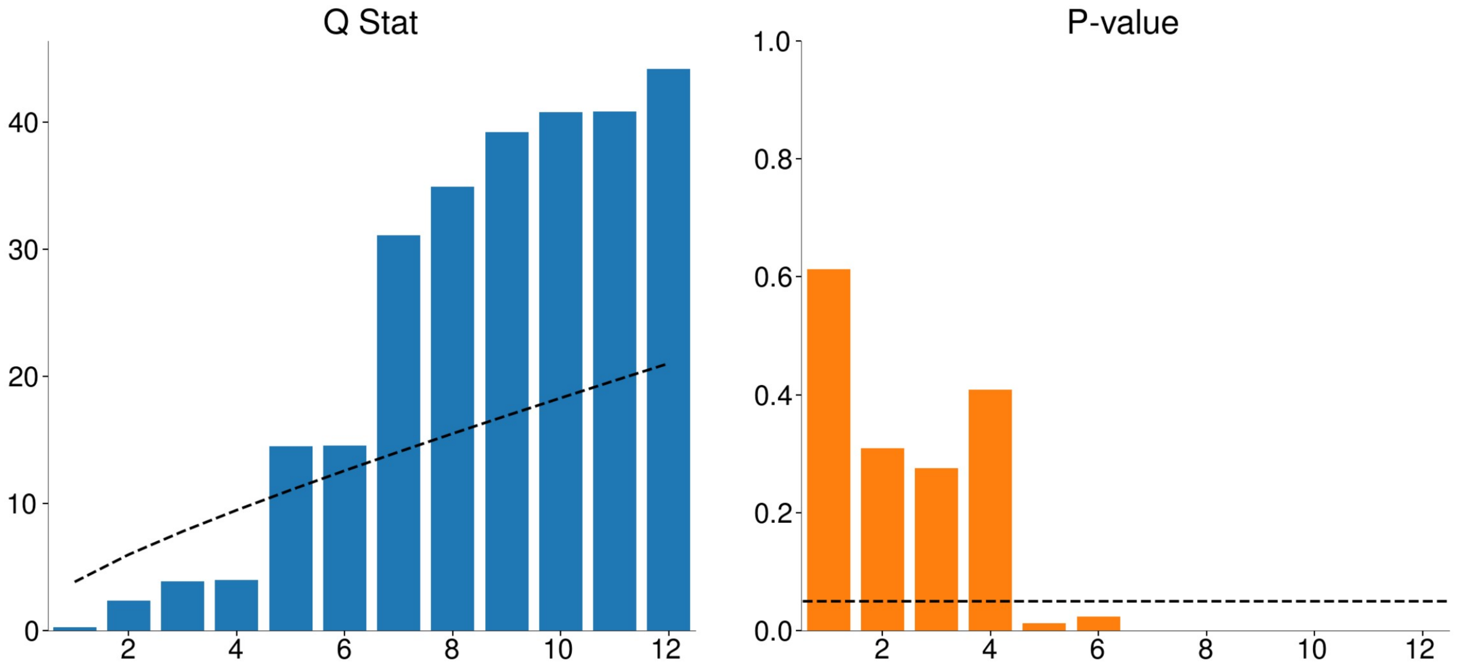
In [59]:

```
res = SARIMAX(default, order=(1, 0, 1)).fit()
resid = res.resid.iloc[1:]
acf_pacf_plot(resid, 24)
```



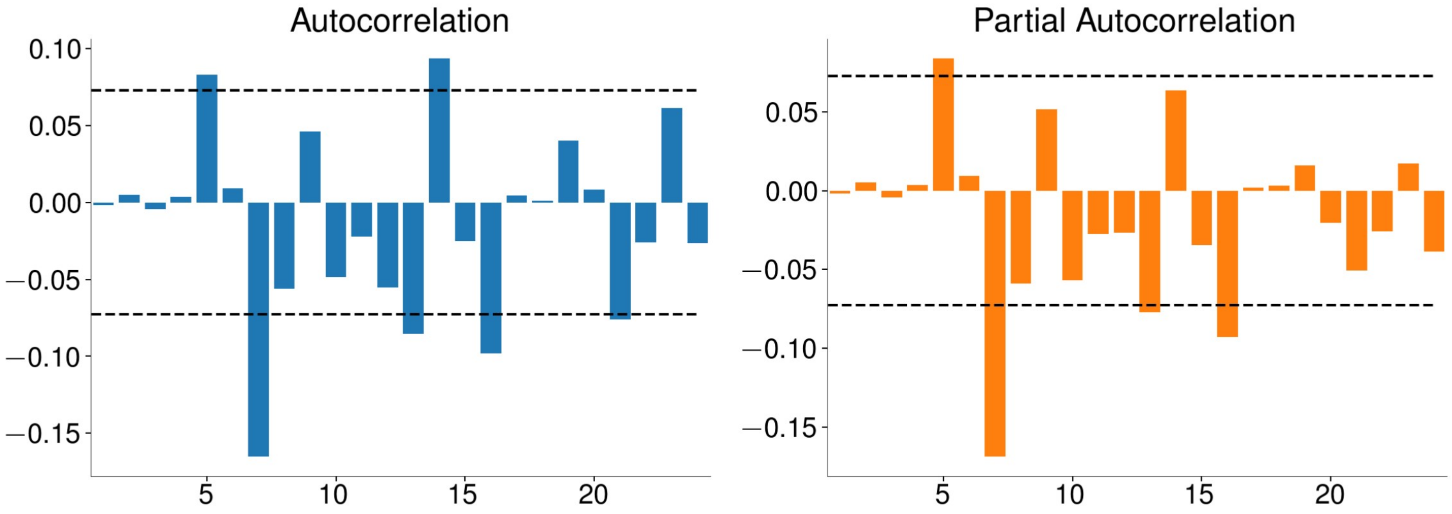
Q Statistic on ARMA(1,1)

In [61]:
```
lb_plot(resid, 12)
```



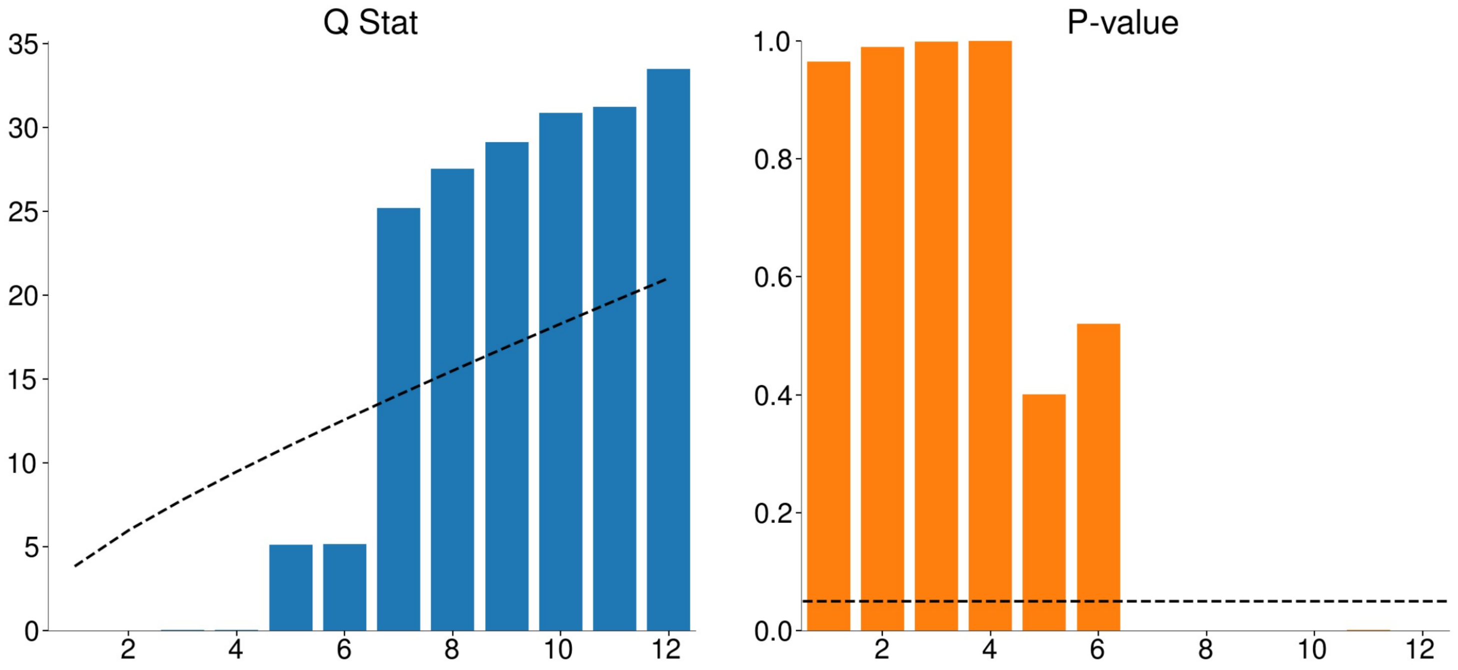
Diagnostics on ARMA(5,1)

In [63]:
```
res = SARIMAX(default, order=(5, 0, 1)).fit()
resid = res.resid.iloc[5:]
acf_pacf_plot(resid, 24)
```



Q Statistic on ARMA(5,1) Residuals

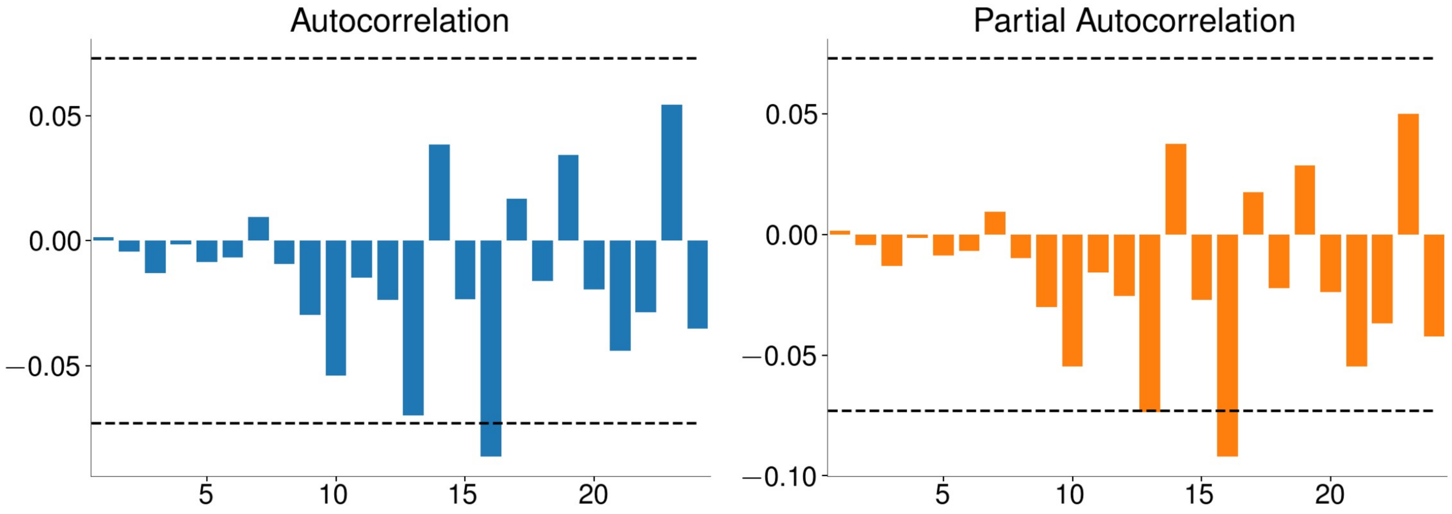
In [65]:
```
lb_plot(resid, 12)
```



Residual Diagnostics on ARMA(10,1)

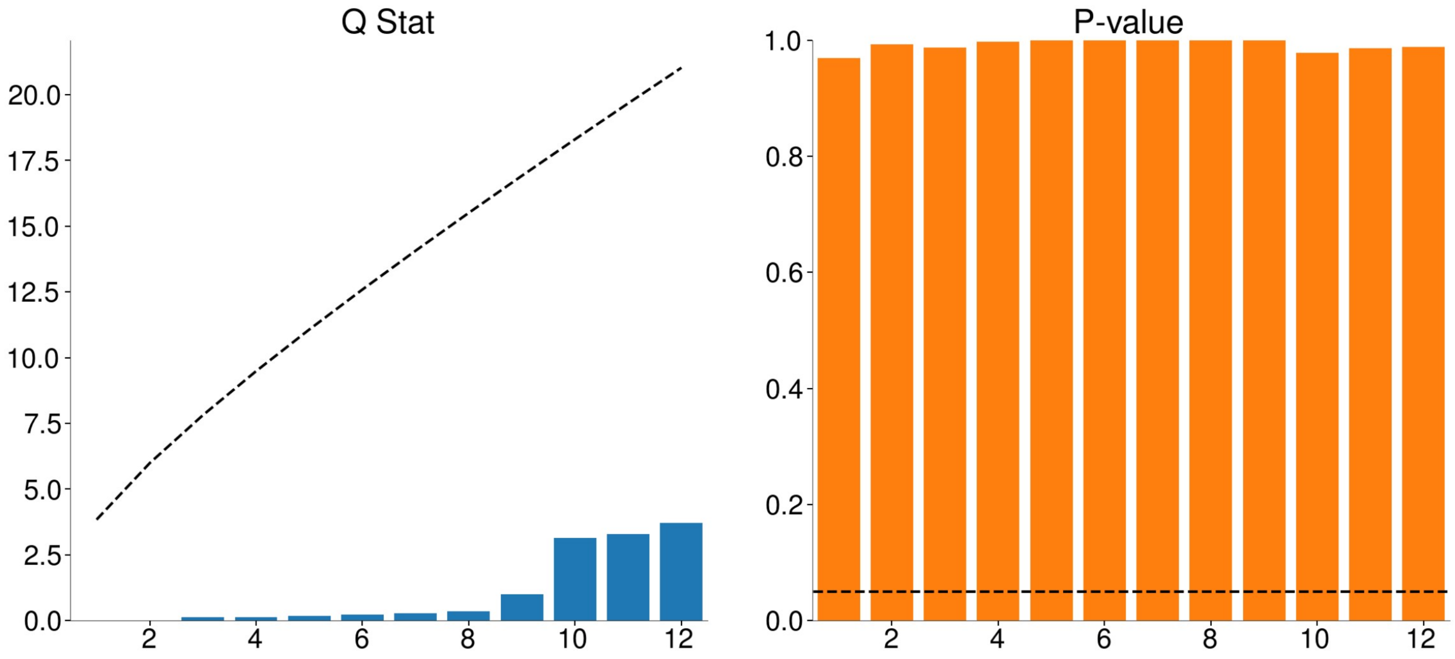
In [67]:

```
res = SARIMAX(default, order=(10, 0, 1)).fit()
resid = res.resid.iloc[10:]
acf_pacf_plot(resid, 24)
```



Q Statistic on ARMA(10,1) Residuals

In [69]:
```
lb_plot(resid, 12)
```



ARMA(10,1) Coefficients

In [70]:
```
summary(res)
```

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.7058	0.202	3.501	0.000	0.311	1.101
ar.L2	0.3550	0.262	1.354	0.176	-0.159	0.869
ar.L3	-0.1445	0.100	-1.448	0.148	-0.340	0.051
ar.L4	0.0532	0.054	0.985	0.324	-0.053	0.159
ar.L5	0.1210	0.030	4.010	0.000	0.062	0.180
ar.L6	-0.0731	0.041	-1.775	0.076	-0.154	0.008
ar.L7	-0.1793	0.049	-3.668	0.000	-0.275	-0.083
ar.L8	0.0536	0.032	1.680	0.093	-0.009	0.116
ar.L9	0.1710	0.036	4.803	0.000	0.101	0.241
ar.L10	-0.0718	0.032	-2.270	0.023	-0.134	-0.010
ma.L1	0.6405	0.197	3.258	0.001	0.255	1.026
sigma2	108.5819	2.292	47.371	0.000	104.089	113.074

IC Selection

Default Premium

- Fit models with P and Q on a grid
- Select smalled AIC or BIC
  - BIC always selects a (weakly) smaller model

In [73]:
```
aic.idxmin()
```

Out[73]:
```
(9, 2)
```

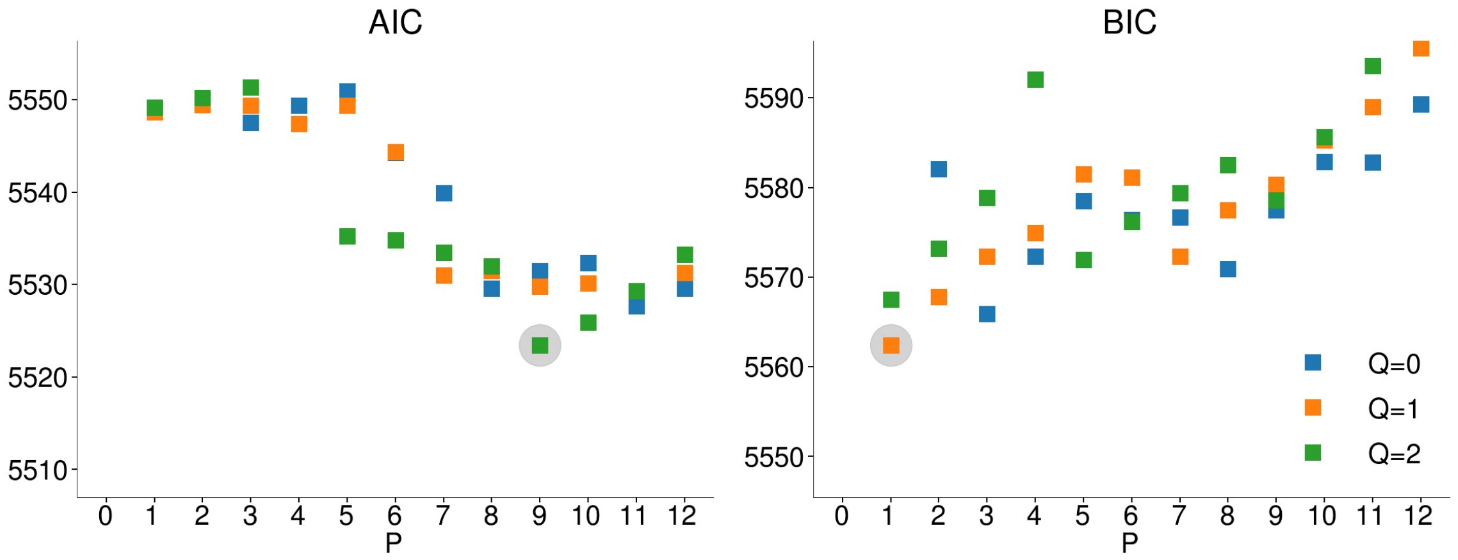
In [74]:
```
bic.idxmin()
```

Out[74]:
```
(1, 1)
```

IC Selection

Default Premium

In [76]:
```
ic_plot(aic, bic)
```



IC Selection

US Construction Worker Growth

In [78]:
```
aic.idxmin()
```

Out[78]:
```
(2, 1)
```

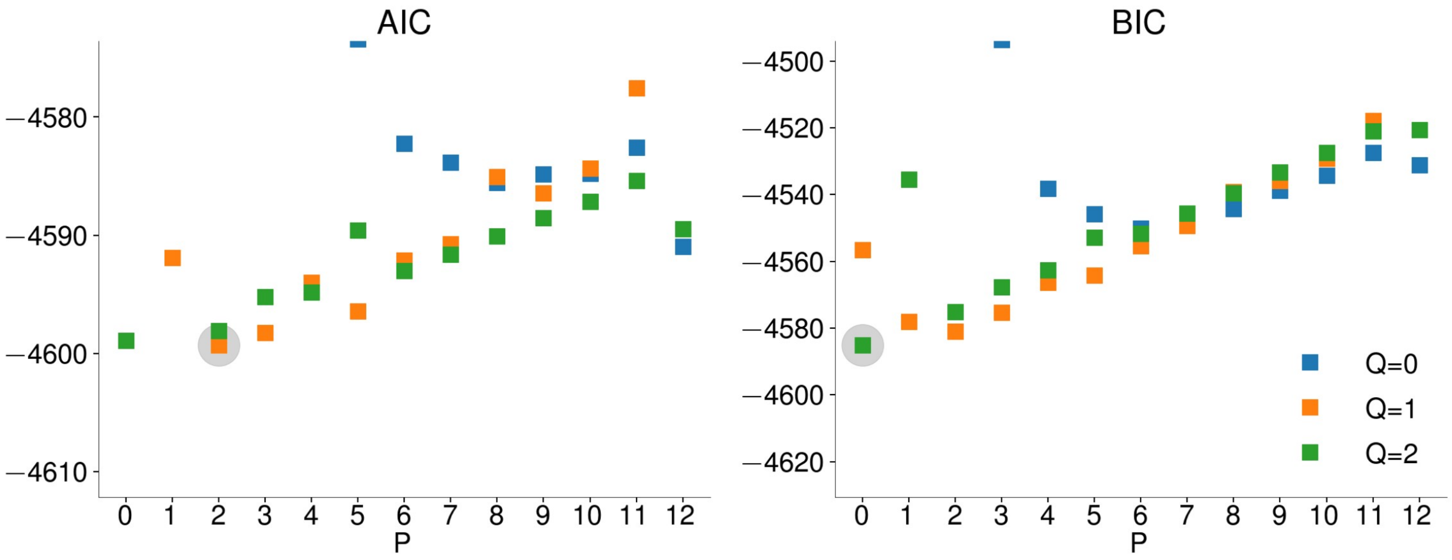
In [79]:
```
bic.idxmin()
```

Out[79]:
```
(0, 2)
```

IC Selection

US Construction Worker Growth

```
In [80]: ic_plot(aic, bic)
```



Next Week

- Forecasting
- Seasonality
- Time Trends
- Unit roots and random walks